UIUCDCS-R-77-887

Math

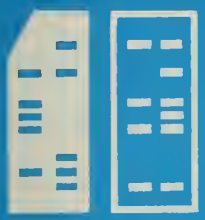
UILU-ENG 77 1743

PROGRAM MANUAL: FOR THE NETTRA SYSTEM

BY

K. C. Hu

August, 1977



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-77-887

PROGRAM MANUAL:
FOR THE NETTRA SYSTEM

by

K.C. Hu

August 1977

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This work was supported in part by the National Science Foundation under Grant No. NSF DCR 73-03421.

ABSTRACT

This is the program manual for the NETTRA system which can design near-optimal, multiple-output, multi-level and loop-free NOR(NAND) networks under fan-in/fan-out restrictions and/or level restriction. Given function(s) may be completely or incompletely specified and both complemented and uncomplemented external variables are permitted as inputs. The user can specify the control sequence (the types of the initial network methods and the types and the order of the transduction procedures to be applied) to solve his problem. Besides, four control sequences are provided for the users who are not interested in the details of how to specify the control sequence. Facilities for treating unfinished jobs due to the expiration of computation time are also provided by the system.

ACKNOWLEDGMENT

The author is grateful for valuable discussions with Professor S. Muroga, H.C. Lai, J. Culliney and, in particular, Professor Muroga's reading of the manuscript.

The nice typing job done by Mrs. R. Taylor is also acknowledged.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION OF THE NETTRA-SYSTEM

The NETTRA system(NETwork TRAnsduction system) is an effective tool for designing near-optimal, multiple-output, multiple-level and loop-free NOR(NAND)-gates networks under fan-in/fan-out restrictions and level restriction.

"Transduction" means "transformation" and "reduction". The purpose of network transductions is to reduce the cost of a network designed for a certain function (or functions) or to reconfigure (or transform) the network in such a way as to allow another transduction to eventually accomplish the cost reduction. This cost, C, of a network is formally defined as $A \times R + B \times I$, where R is the number of gates, and I is the number of connections in the network; A and B are arbitrary non-negative weights. Different combinations of A and B give different cost criterion for the transduction.

1-1 Outline of the NETTRA system

The NETTRA system can treat the following four types of problems:

(1) Find near-optimal networks for the given function(s) under no fan-in/fan-out restrictions or level restriction.

(2) Find near-optimal networks for the given function(s) under only fan-in/fan-out restrictions.

(3) Find near-optimal networks for the given function(s) under only level restriction.

(4) Find near-optimal networks for the given function(s) under both fan-in/fan-out restrictions and level restriction.

Here by "under fan-in/fan-out restriction" we mean that the maximum fan-in of each gate and/or the maximum fan-out of each external variable or each gate in the network are restricted; by "under level restriction" we mean that the maximum number of levels in the network is restricted.

For each type of problem, the NETTRA system first produces the initial networks for the given functions by using some conventinal design methods [1]. The initial networks usually have many redundant gates and/or connections, although they can be produced in a very short computation time. So appropriate transformation and/or transduction procedures are applied by the NETTRA system to simplify the initial networks.

For type (1) problems, the NETTRA system processes networks according to the flowchart shown in Fig. 1.1-1. In Fig. 1.1-1, after the initial network has been derived, the transduction procedures are applied, without considering fan-in/fan-out restrictions, to simplify the initial networks as much as possible. For type (2) problems, the corresponding flowchart is shown in Fig. 1.1-2. After obtaining the initial networks, the transduction procedures are applied to simplify the initial network first without considering the fan-in/fan-out restrictions. The simplified network has lower cost, but it may not satisfy the given fan-in/fan-out restrictions. Therefore the transformation procedure is applied to transform the network into fan-in/fan-out restricted form. Then the transduction procedures are applied again, considering the fan-in/fan-out restriction, to simplify the network as much as possible. The resulting network obtained at this point is a feasible solution for the given problem, i.e., a network which satisfies the given restrictions. But we can apply the non-fan-in/non-

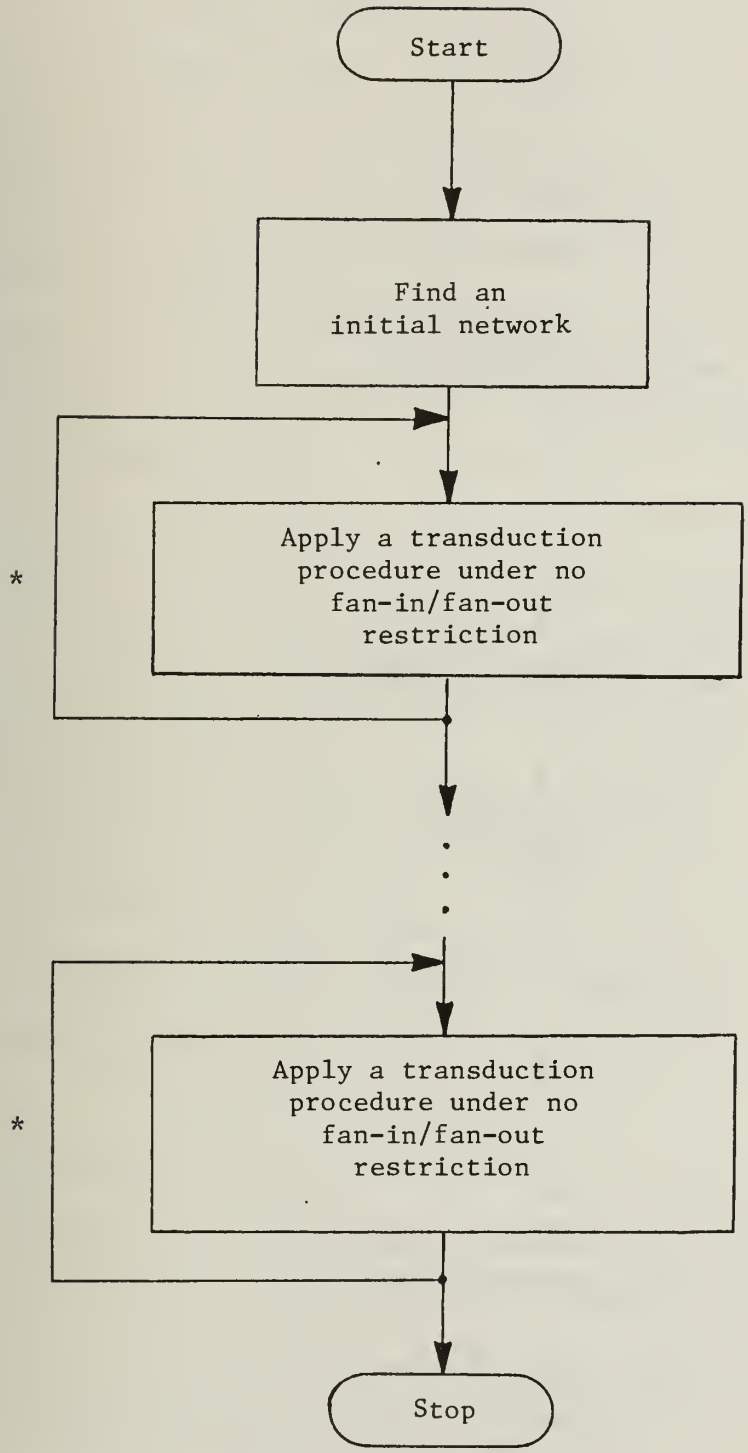


Fig. 1.1-1 Flowchart for solving type (1) problems.

*These loops will be executed repeatedly until there is no further improvement in the network cost.

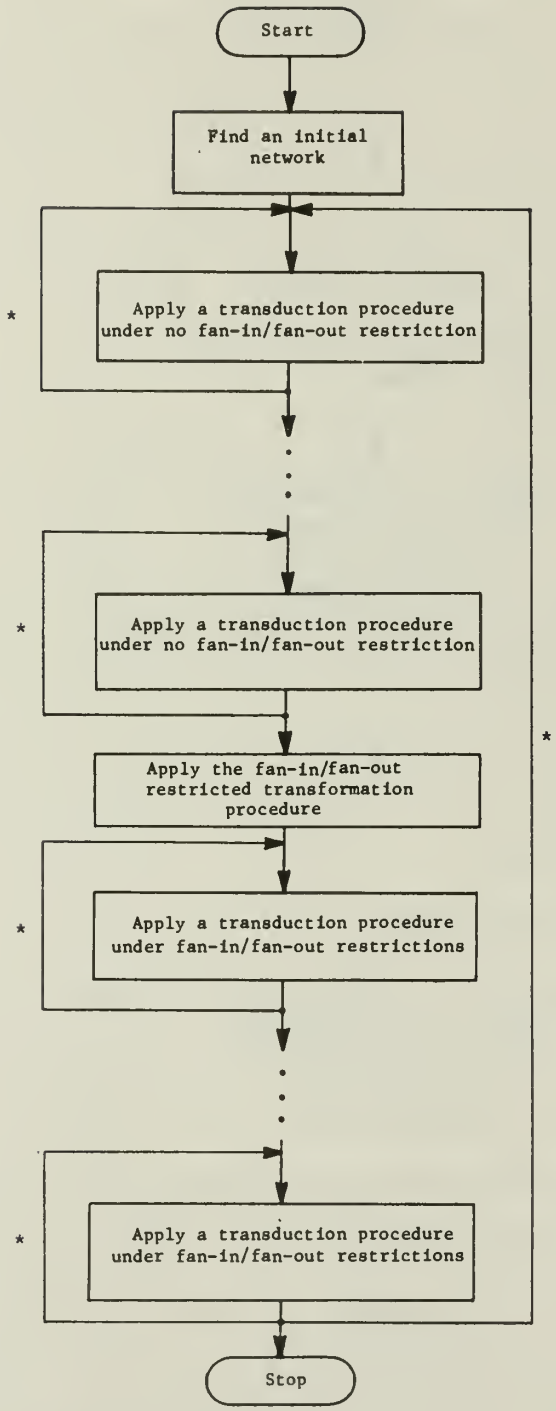


Fig. 1.1-2 Flowchart for solving type (2) problems

*These loops can be executed repeatedly until there is no further improvement in the cost.

fan-out restricted transductions, fan-in/fan-out restricted transformation, and fan-in/fan-out restricted transductions repeatedly in order to get a more simplified network. The flowchart for type (3) problems is shown in Fig. 1.1-3. First the NETTRA system finds an initial network whose number of levels is less than or equal to the specified limit, and then the transduction procedures are applied, considering the level restriction, to simplify the initial network as much as possible. The flowchart for type (4) problems is shown in Fig. 1.1-4. The NETTRA system tries to find an initial network which satisifies the level restriction and has as less fan-in/fan-out problems* as possible. The transduction procedures will, then, be applied to simplify the initial network, considering the fan-in/fan-out restrictions and level restriction. During the transductions, the number of levels of the network will never exceed the limit, and some fan-in/fan-out problems may be solved. If all fan-in/fan-out problems can be solved, then a feasible network has been obtained. If the transductions cannot solve all fan-in/fan-out problems, then no feasible network is obtained by the approach shown in Fig. 1.1-4 even if there do exist feasible solutions for the given problem†.

In Fig. 1.1-1 through Fig. 1. 1-4, those loops marked with asterisks can be executed as many times as the users wish. Besides, the users can specify the types of initial network methods and the types and the order of the transduction procedures to be applied to solve the problems according to

---

*If the fan-in of a gate exceeds the limit, then we say there is a fan-in problem at that gate. The fan-out problem of a gate or an external variable is similarly defined.

†There may not exist feasible networks for a given problem if both fan-in/fan-out and level restrictions both are imposed.

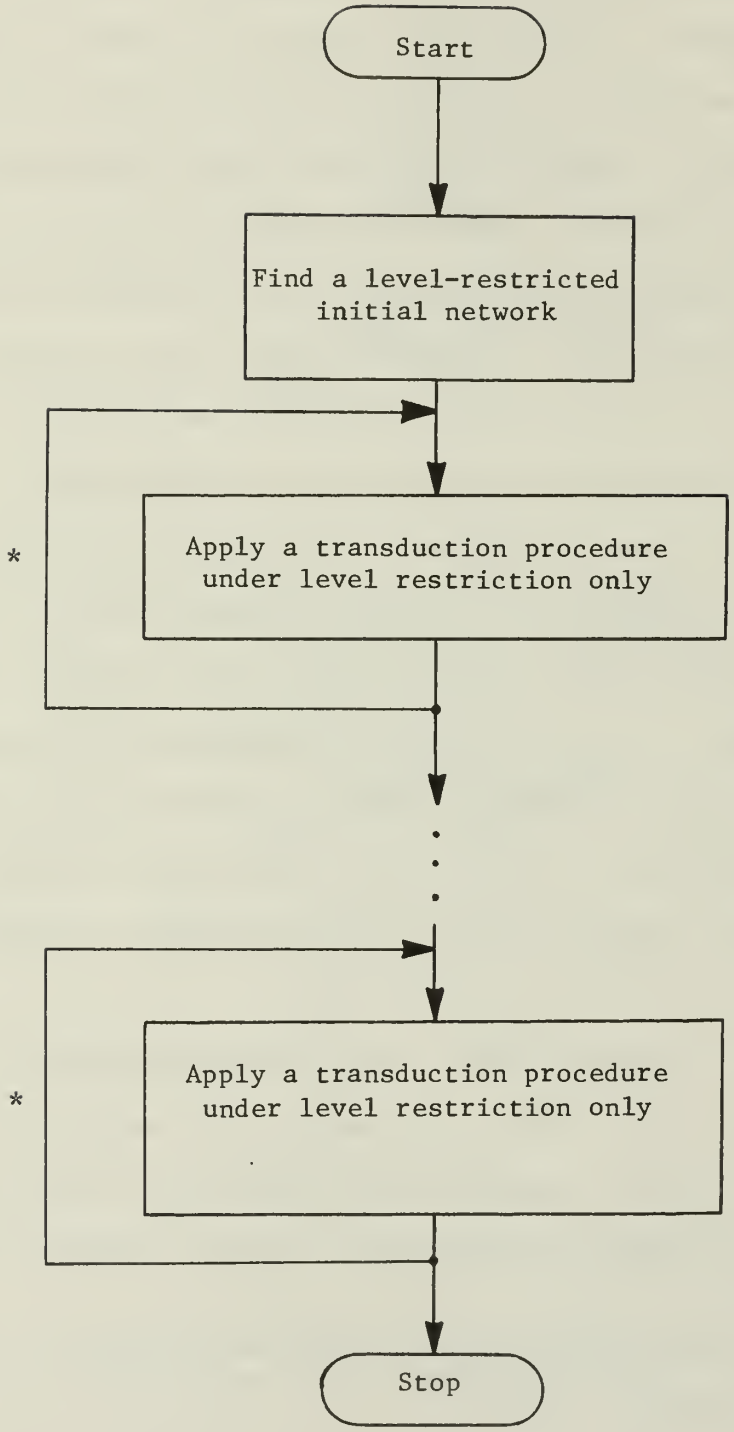


Fig. 1.1-3 Flowchart for solving type (3) problems.

*These loops can be executed repeatedly until there is no further improvement in the cost.

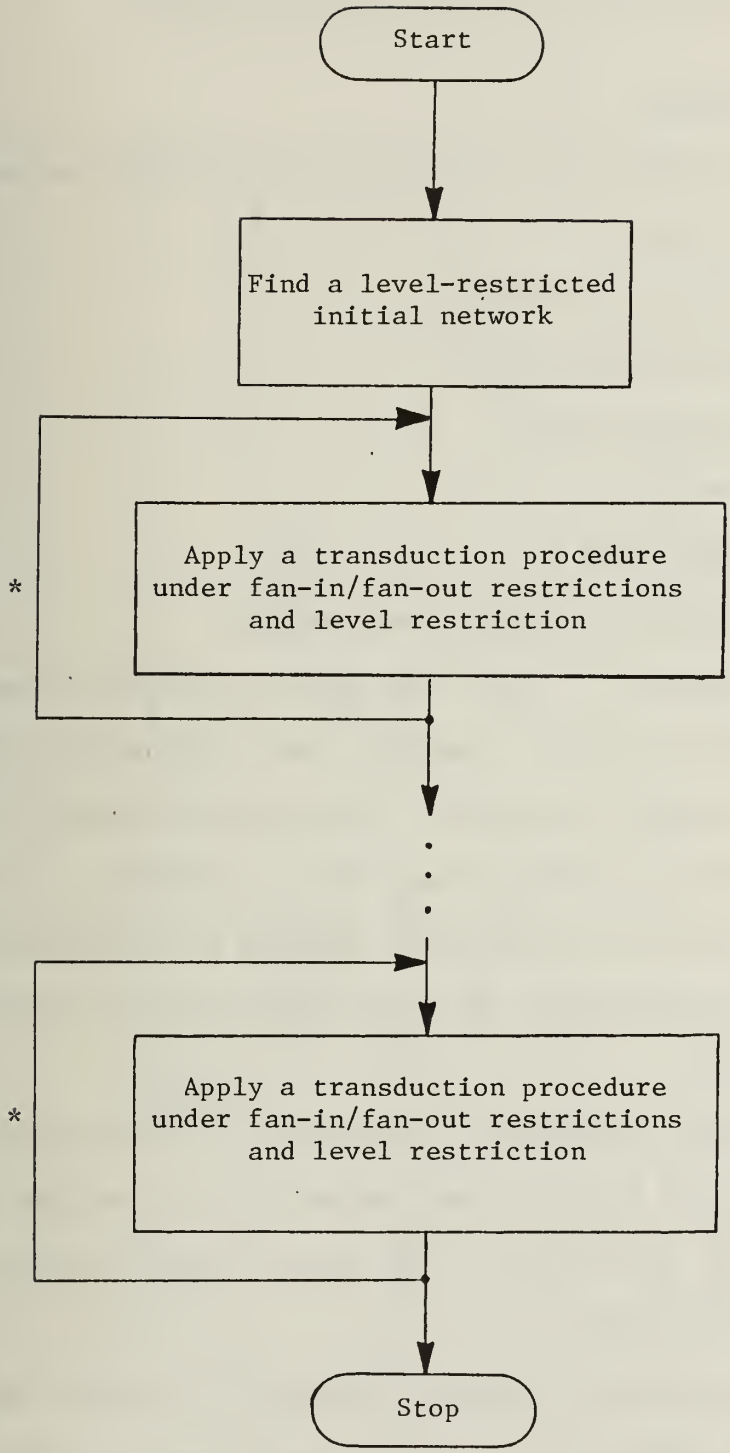


Fig. 1.1-4 Flowchart for solving type (4) problems.

*These loops can be executed repeatedly until there is no further improvement in the network cost.

their specific purposes. The general tendency of effectiveness and efficiency of different transduction procedures and the characteristics of different initial network methods will be given in the following sections.

1-2 Initial network methods

Six different methods are implemented in the NETTRA system to produce initial networks. They are:

1. Universal NOR network method.
2. THREE-level network method.
3. Branch-and-bound method.
4. Tison's method.
5. Gimpel's algorithm.
6. Level-restricted initial network method.

Among them, methods 1, 2 and 5 can produce networks with uncomplemented external variables as inputs. Methods 2 and 5 produce only three-level networks, and method 4 produces two-level or three-level networks depending on whether both uncomplemented and complemented external variables or only uncomplemented external variables are available. Method 6 is for designing the level-restricted initial network when both fan-in/fan-out restrictions and level restriction are imposed.

Table 1.2-1 gives the mnemonic symbols for six initial network methods; it also shows what initial network methods can be used for what type of problems (mentioned in section 1.1). The mnemonic symbols will be convenient for setting up data cards.

In [1] many experiments have been conducted to find out the influence of initial network methods upon the final results and to compare the computation time that different initial network methods spend. Table 1.2-2

Table 1.2-1 Mnemonic symbols for initial network methods and the relationship between the initial network methods and 4 types of problems (mentioned in section 1.1).

INITIAL NETWORK METHODS	MNEMONIC SYMBOLS	TYPE OF PROBLEMS			
		1	2	3	4
Universal NOR network method	UNIV	X	X		
Three-level network method	THRL	X	X	X*	
Branch-and-bound method	BANB	X	X		
Tison's method	TISN	X	X	X	
Gimpel's algorithm	TANT	X	X	X*	
Level-restricted initial network method	TLEV		X†		X

*These initial network methods can be used only when the level limit is greater than or equal to 3.

†The level-restricted initial network method can also be used to generate a fan-in/fan-out restricted initial network by setting the maximum level limit as 100.

gives the average computation time spent by five initial network methods for finding initial networks for ten 5-variable functions. The initial network method based on Gimpel's algorithm is relatively more time-consuming than others. Also since the level-restricted initial network method is designed for special purposes, it is not compared with others in Table 1.2-2; but the time spent by this method is usually a little more than the time spent by Tison's method.

The statistics shown in [1] also indicate that the final results (or the costs of the resultant networks) do not have direct relationships with the costs of the initial networks, i.e., starting from an initial network with lower cost does not guarantee that a better network can be obtained after applying the transduction and the transformation procedures.

Table 1.2-2 Comparison of average computation times for initial network methods (ten 5-variable functions are used).

UNIV	THRL	BANDB	TISN	TANT
4.4 cs*	7.5 cs	77.8 cs	29.6 cs	> 1 min†

*cs = centisecond †min = minute

Usually the initial networks obtained by Tison's method have lower costs than those obtained by other methods. But it has been seen that starting from the initial networks obtained by the branch-and-bound method, better final results can usually be obtained. This is because the initial networks obtained by the branch-and-bound method are usually multiple-level networks, and hence are more suitable for the transduction procedures to reconfigure. The initial networks obtained by Tison's method are restricted to be two-level or three-level, and hence are more difficult to reconfigure even though they have lower

costs. The initial networks obtained by three-level network method are also restricted to be two-level or three-level; since they usually have higher costs than the initial networks obtained by Tison's method, the corresponding final results are worse. The initial networks obtained by the universal network method are multiple-level, but they usually contain too many gates and connections and hence are more difficult to simplify.

1-3 Transformation and transduction procedures

In the NETTRA system, there is only one transformation procedure which can transform the network from non-fan-in/non-fan-out restricted form into fan-in/fan-out restricted form. This transformation procedure is used for solving problems under only fan-in/fan-out restrictions, i.e., for solving type (2) problems. The mnemonic symbol for this transformation procedure is JEFF.

The transduction procedures can be classified into the following five groups according to their characteristics and capabilities:

1. Pruning procedures.
2. Procedures based on gate merging.
3. Procedures based on gate substitution.
4. Procedures based on connectable and disconnectable functions.
5. Procedures based on error-compensation.

Table 1.3-1 gives the names and the mnemonic symbols of the central FORTRAN subroutines for realizing each group of procedures. The mnemonic symbols are originated from the transduction programs NETTRA-PG1, -Pg2, -P1, -P2, -G1, -G2, -G3, -G4, and -E1 [1]. They are convenient for specifying input data cards.

Table 1.3-1 Central subroutines and their mnemonic symbols of the transduction procedures.

TYPE OF TRANSDUCTION PROCEDURES	NAMES OF CORRESPONDING CENTRAL SUBROUTINES	MNEMONIC SYMBOLS
PRUNING PROCEDURES	MINI2	NPGM
	RDTCNT	NTP1
	PROCI	NTP2
PROCEDURES BASED ON GATE SUBSTITUTION	SUBSTI	NPGS
	PROCV	NTG4
PROCEDURES BASED ON GATE MERGING	GTMERG	NTG3
PROCEDURES BASED ON CONNECTABLE AND DISCONNECTABLE FUNCTIONS	PRIIFF	NTG1
	PROCIV	NTG2
PROCEDURES BASED ON ERROR-COMPENSATION	PROCCE	NTE1

Many experiments have been made to find out the effectiveness and the efficiency of the transduction procedures. The result is shown in Fig. 1.3-1; the detailed comparisons are given in [1]. Notice that all transduction procedures except NTG4 can do transductions without violating the fan-in/fan-out restrictions and/or level restriction if the network before applying transductions is already fan-in/fan-out restricted and/or level-restrict. NTG4 is not included in Fig. 1.3-1, but usually it is less effective but mor time-consuming than NPGS in doing transductions under no fan-in/fan-out restrictions and no level restriction [1].

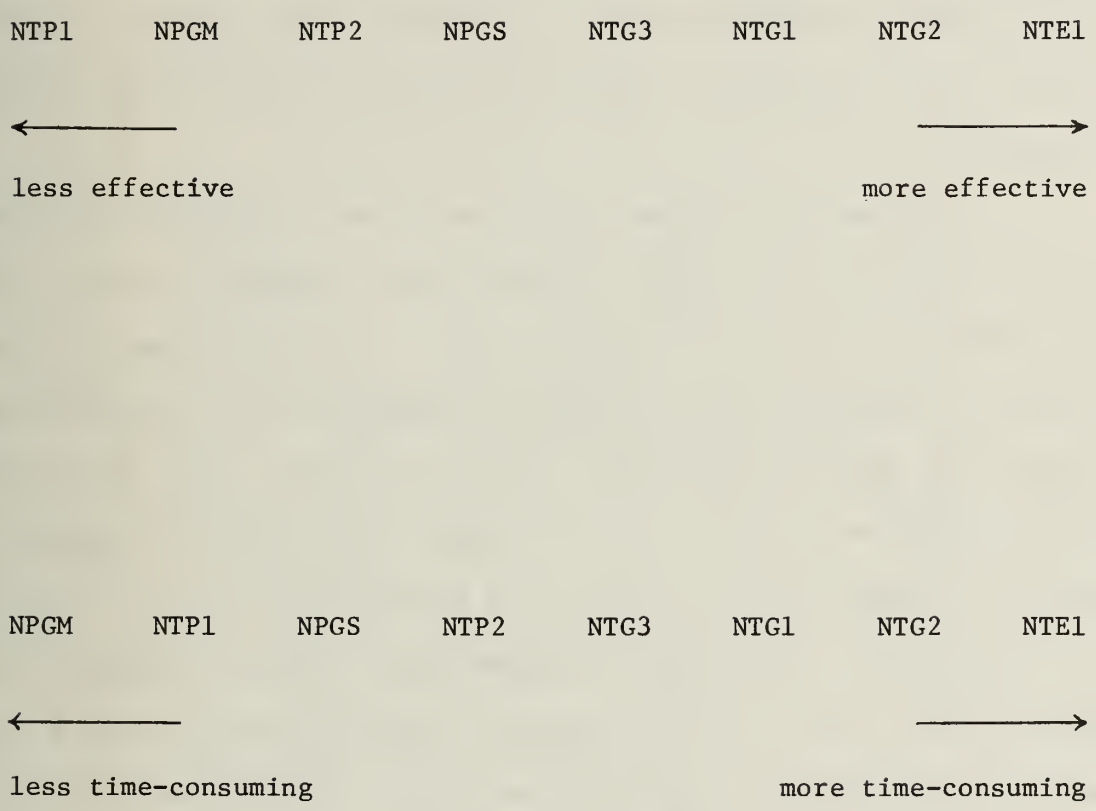


Fig. 1.3-1 General tendency of the effectiveness and the efficiency of the transduction procedures. (NTG4 is not included.)

1-4 Control sequences

A control sequence consists of one or more initial network methods and a TT-sequence (Transformation and Transduction sequence), where each TT-sequence consists of the types, the order and the numbers of execution times of the transduction procedures to be applied. Starting from the initial network obtained by one of the initial network methods, the TT-sequence will be applied to simplify the network as many times as users specified.

The users can design control sequences by choosing appropriate initial network methods and the transduction procedures according to Table 1.2-1, Fig. 1.3-1 and the statistics in [1]. Besides, for users' convenience four control sequences, OPTION 1 through OPTION 4, are provided internally for solving type (2) and type (4) problems. Since type (1) and type (3) problems are special cases of type (2) and type (4) problems, respectively, these control sequences can be used to solve type (1) and type (3) problems too. The contents of these built-in control sequences are shown in Table 1.4-1. Control sequences OPTION 1 through OPTION 3 are for designing fan-in/fan-out restricted networks. The difference among them are that OPTION 1 aims at finding networks with very good costs, usually taking more computation time, OPTION 2 aims at finding networks with reasonably good costs in a reasonably short time, and OPTION 3 tries to find networks in a very short time, though their costs may not be good. Control sequence OPTION 4 is for designing fan-in/fan-out restricted and level-restricted networks. It does not guarantee that feasible networks can always be obtained.

If the users are not interested in setting up their own control sequences, they can simply select one of four built-in control sequences to solve their problems.

Table 1.4-1 Contents of four built-in control sequences

CONTROL SEQUENCE TYPE	INITIAL NETWORK METHODS	Non-fan-in/Non-fan-out restricted transduction step	Fan-in/fan-out restricted transformation step	Fan-in/fan-out restricted transduction step
OPTION 1	UNIV THRL BANB	NTG3 ∞*, NTG1 ∞	JEFF	NTG2 ∞, NTE1 ∞
OPTION 2	BANB	NTG3 ∞, NTG1 ∞	JEFF	NTE1 ∞
OPTION 3	UNIV	MPGM ∞, NTG1 ∞	JEFF	NTG1 ∞
OPTION 4	TLEV			NPGS ∞, NTG3 ∞, NTG1 ∞, NTG2 ∞, NTE1 ∞

*∞ means that the selected transduction procedure is going to be applied repeatedly until there is no improvement in the network cost.

2. SET-UP OF INPUT DATA

For each separate run, a set of problems may be submitted. Input data cards consists of six types of cards:

(i) < specification card >

(ii) < heading card >

(iii) < problem parameter card >

(iv) < control sequence card > s

(v) < output function card > s

(vi) < connection description card > s

The user must prepare the input cards (i), (ii), (iii), (iv) and (v) for the first problem. Type (vi) cards must be prepared if the initial network is to be read in explicitly. For later problems, type (i) cards must not be prepared and other types of cards must or must not be prepared depending on the information of the < specification card >.

The following is the formats of these cards.

(i) < specification card >: This is the first card of the first problem. It contains several parameters which tell whether the following problems need types (ii), (iii), (iv), and (v) cards. There are 6 fields on this card.

cols. 1~4: An integer, HEAD, which is right justified. This integer may be zero or nonzero. If it is zero, then the following problems do not have heading cards. This is the case when the following problems have the same heading information as the first problem. If HEAD is nonzero, then a heading card must be prepared for each of the following problems.

cols. 5∿8: An integer, FUNC, which is right justified. It may be zero or nonzero. When it is zero, the following problems will have the same output function(s) as the first problem. This may happen when we want to realize the same function(s) but under different constraints or applying different control sequences. If FUNC is nonzero, then we must prepare < output function card >s.

cols. 9∿12: An integer, PARM, which is right justified. When it is zero, no < problem parameter card >s are required for the following problems, i.e., the following problems use the parameter values specified for the first problem. When PARM is nonzero, a < problem parameter card > must appear in each of the following problems.

cols. 13∿16: An integer, SEQC, which is right justified. When it is zero, the < control sequence card >s specified for the first problem will be used for the following problems. When SEQC is nonzero, the < control sequence card >s must be specified for each of the following problems.

cols. 17∿20: An integer, PUNC, which is right justified. It may be zero or nonzero. If it is nonzero, then the best results for each problem will be punched on the cards*. The format of the punched cards is the same as that of the < connection-description card >s. If PUNC is zero, then no cards will be punched.

cols. 21∿24: An integer, TLIM, which is also right justified. It specifies the limit of computation time that the user likes to spend to solve each problem. If any problem cannot be finished in the specified time limit, then intermediate results will be punched so that the user can continue running this problem next time. After punching the intermediate

---

*These cards may be used as input data for the CALCOMP program for drawing the network.

results, the next problem will be read in and processed.

TLIM is implicitly expressed in seconds, and it may have the maximum value 9999 seconds (or 2 hours, 46 minutes and 39 seconds). Recommended TLIM for a typical 3-, 4- or 5-variable problem is 2 minutes, 5 minutes or 10 minutes, respectively. Most of the 3-, 4- or 5-variable problems can be finished within the time limit recommended above.

The < specification card > is helpful for saving the preparation of some data cards when a set of problems having the same information are submitted in one run.

(ii) < heading card >: This card may contain any alphanumeric information in column 1-80. It is for the identification of each problem and no information on this card will be used in the actual computation.

(iii) < problem parameter card >: This card specifies the nature of each problem the user wants to solve. There are fourteen fields in this card. Each field is composed of characters or numerals.

cols. 1∿4: An integer, N, which is right justified. It specifies the number n of external variables. Be sure to punch n, rather than 2 × n, for N in the case that both complemented and uncomplemented external variables are available.

cols. 5∿8: An integer, M, which is right justified. This parameter specifies the number m of output functions for the current problem.

cols. 9∿12: An integer, A, which is right justified. The number A is the value of the non-negative weight for the number of gates in the cost function. Table 2-1 gives the typical combinations of values A and B for different network reduction problems.

cols. 13∿16: An integer, B, which is right justified. The number B is the value of the non-negative weight for the number of connections in

Network Reduction Problem	Values of A and B
reducing the number of gates only.	A = 1 and B = 0
reducing the number of gates primarily, then reducing the number of connections secondarily*.	A = 1000 and B = 1
reducing the number of connections only.	A = 0 and B = 1
reducing the number of connections primarily, then reducing the number of gates secondarily.	A = 1 and B = 1000
reducing the sum of the number of gates and the number of connections.	A = B = 1

Table 2-1 Typical combinations of values A and B for different network reduction problems.

*Most of the programs in the NETTRA system are oriented toward this reduction problem, so the user will probably find this combination of A and B the most useful.

the cost function (see Table 2-1).

cols. 17∿19: blank.

col. 20: A blank or one of the characters "X" and "Y". The blank or the character X means that only uncomplemented external variables are available as inputs. The character Y means that both complemented and uncomplemented external variables are available.

cols. 21∿24: An integer, TFI, which is right justified. This parameter specifies the maximum number of fan-in a gate may have. The default value is 100* when this field is zero or blank.

cols. 25∿28: An integer, TFO, which is right justified. It specifies the maximum number of fan-out for gates (not output gates). The default value is 100 when this field is zero or blank†.

cols. 29∿32: It specifies the maximum number of fan-out for external variables. The default value is 100 when this field is zero or blank†.

cols. 33∿36: An integer, TFOO, which is right justified. It specifies the maximum number of fan-out for output gates. The default value is 100 when this field is zero or blank†. If the user does not want to have some output gates having connections to other gates, he has to specify a negative number, e.g. -1.

cols. 37∿40: An integer, LREST, which is right justified. It specifies the maximum number of levels that the network may have. The default value is 100 when this field is zero or blank‡.

---

*This is equivalent to solving the problem under no fan-in restrictions.

†These are the cases of no fan-out restrictions for external variables or gates.

‡This is the case when there exists no level restriction.

cols. 41∿44: An integer, POPT1, which is right justified. This parameter may have value 1 or 0. When it is 1, the detailed processes for initial network subroutine TISLEV will be printed. Usually this field is specified as zero or blank so that no intermediate result about TISLEV is printed.

cols. 45∿48: An integer, POPT2, which is right justified. It has the value 1 or 0. When it is 1, the detailed information about the transduction procedures will be printed. Usually, it is left as blank.

cols. 49∿52: An integer, RERUN, which is right justified. This parameter indicates whether the current problem is run for the first time or not. If the current problem is run for the first time, then RERUN has to be set the value zero. For the problem which is not finished last time, the RERUN field on the parameter card in the punched deck is specified as 1.

cols 53∿56: An integer, NEPMAX, which is right justified. This parameter specifies the maximum number of error positions permitted in the transduction procedures based on error-compensation. Usually this field is left as blank even if error-compensation transduction is involved in the problem. The default value is $2^{(n-1)}$, where n is the number of external variables.

(iv) < control sequence card >s: The control sequence cards specify the types of initial network methods and the types, the order and the number of execution times of the transduction procedures to be applied for solving the current problem.

The following four keywords denote four built-in control sequences. They are OPT1, OPT2, OPT3 and OPT4, which correspond to control sequences

OPTION 1, OPTION 2, OPTION 3 and OPTION 4, respectively. The user need only specify one of these keywords on the control sequence card. Since free format is used to specify the control sequence cards, this keyword can appear anywhere on the card as the first four nonblank characters.

For the user who intends to specify his own control sequence, the following descriptions must be followed.

Twenty-two keywords are available for the specification of control sequences. Each keyword consists of four characters. Table 2-2 shows these keywords and their corresponding meanings. Since free format is used, one or more blanks must be used to separate two keywords, a keyword and a numeral, or a keyword and a character (this will be clarified soon).

INTP is the first four nonblank characters that have to appear on the control sequence cards. It works as the start of a control sequence. The keywords for the selected initial network methods must be specified one by one after INTP. In Table 2-2, seven keywords are used for specifying initial network methods. Among them, EXNT is used for reading in an initial network prepared by the user. The meanings for the six keywords except EXNT are introduced in Table 1.2-1.

One of the keywords TDTP, JEFF, FLTP and NOIT must appear following the keyword specifying one of the initial network methods. Whenever TDTP appears, the transduction procedures without fan-in/fan-out restrictions and level restriction will be applied, and the keywords for the selected transduction procedures and the number* of execution times must

---

*It can be a positive integer or the character #. When # is used, the selected transduction procedure will be applied repeatedly until there is no further improvement in the network cost.

Table 2-2 Keywords for specifying control sequence cards

KEYWORD	MEANING
INTP	Types of initial network methods
UNIV	INITIAL
THRL	
BANB	
TISN	NETWORK
TANT	
TLEV	METHODS
EXNT*	
TDTP	No fan-in/fan-out restricted transduction step
JEFF	Fan-in/fan-out restricted transformation step
FLTP	Fan-in/fan-out restricted and/or level-restricted transduction step
NOIT	Number of iterations in the application of a TT-sequence
STOP	End of the control sequence
NPGM	TRANSDUCTION PROCEDURES
NPGS	
NTP1	
NTP2	
NTG1	
NTG2	
NTG3	
NTG4	
NTE1	

*This keyword is for reading in an initial network prepared by the user.

appear following TDTP. Whenever FLTP appears, the transduction procedures will be applied under fan-in/fan-out restrictions and/or level restriction, and the keywords for the selected transduction procedures and the number* of execution times must appear following FLTP. It should be noticed that NTG4 cannot be used for fan-in/fan-out restricted or level-restricted transduction. Whenever JEFF appears, the fan-in/fan-out restricted transformation procedure will be applied. Whenever NOIT appears, the contents of the TT-sequence have been specified; so the number of iterations of the TT-sequence must be given after NOIT. This number can be a positive integer or the character #. When # is used, the TT-sequence will be applied repeatedly until there is no further improvement in the network cost.

Notice that keywords JEFF and FLTP can never appear before the keyword TDTP if all of them appear on the card, and keyword FLTP cannot appear before the keyword JEFF if they both appear on the card. Table 2-3 gives the order of keywords INTP, TDTP, JEFF, FLTP and NOIT for different types of problems.

Table 2-3 Order of keywords for different types of problems

Problems Type	Keywords				
	INTP	TDTP	JEFF	FLTP	NOIT
(1)	1	2			3
(2)	1	2	3	4	5
(3)	1			2	3
(4)	1			2	3

*It can be a positive integer or the character #. When # is used, the selected transduction procedure will be applied repeatedly until there is no further improvement in the network cost.

The following is an example for a type (2) problem.

Example Suppose three initial network methods UNIV, THRL and BANB will be used to generate initial networks and the TT-sequence contains the following information:

1. Apply NPGS 3 times, NTG3 2 times, under no fan-in/ fan-out restrictions.
2. Apply the transformation procedure JEFF.
3. Apply NTG1 repeatedly, and then apply NTE1 2 times under fan-in/fan-out restrictions.
4. The TT-sequence is to be executed repeatedly.

The specification of the control sequence for this problem is

INTP UNIV THRL BANB TDTP NPGS 3 NTG3 2 JEFF FLTP NTG1 # NTE1 2 NOIT # STOP

The shortest length of this sequence is 74 characters, so one card is enough.

In any control sequence, the keyword STOP is used to indicate the end of the specification.

(v) < output function card >s: The m < output function card >s specify the set of m output functions to be realized simultaneously. Each function is expressed in the truth table form. Each card contains the values of one output function, starting from column 1 of the card (i.e., left justified). The maximum number of binary digits for each function is limited to 32.

(vi) < connection-description card >s: The < connection-description card >s are used for inputting the configuration of a network prepared by the user. Each of these cards is divided into 40 fields of 2 columns each (i.e., columns 1∿2, 3∿4, ... 77∿78, 79∿80). Beginning with the first

field of the first card, continuing through the succeeding fields of that card and through the fields of as many additional cards as necessary, specify the network configuration right justified in their respective fields.

The network configuration is expressed according to the following description.

1. Label each gate of the network uniquely by assigning to it one of the integers 1, 2, ..., R, such that the output gates receive the labels 1, 2, ..., m, where R is the number of gates in the network.

2. Assign the names X1, X2, ..., XN to the uncomplemented external variables x_1, x_2, ..., x_n and the names Y1, Y2, ..., YN to the complemented external variables $\bar{x}_1$, $\bar{x}_2$, ..., $\bar{x}_n$, respectively.

3. For each gate I, $1 \leq I \leq R$, which has some external variables and the outputs of some other gates as inputs, specify the numeric label of gate I in one field, specify the alphanumeric labels of external variables and the numeric labels of other gates in the following fields continuously, and finally leave one field blank (two blank columns) to indicate the end of the specification of the connection configuration for this gate.

After all gates have been specified in a manner explained above, one field must be filled with two asterisks to indicate the end of the connection-description cards.

An example is shown below. In Fig. 2-1(a), a network with 6 gates (R=6) is given. Both complemented and uncomplemented external variables are available. The specification for the connection configuration is shown in Fig. 2-1(b)

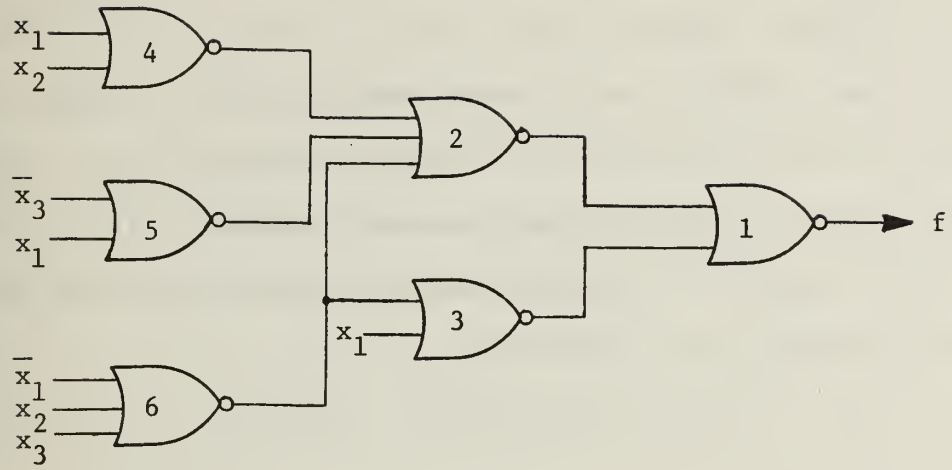


Fig. 2-1(a) The given network

Fig. 2-1(a) The given network

col. 1 col. 56

```
_1_2_3___2_4_5_6___3X1_6___4X1X2___5Y3X1___6Y1X2X3__**
```

Fig. 2-1(b) The card to specify the connection configuration for the network in (a).

Notice that in specifying the input connection configuration for each gate, there is no restriction that external variables should appear earlier than other gates, and also there is no restriction that the gates should appear in order according to their numeric labels.

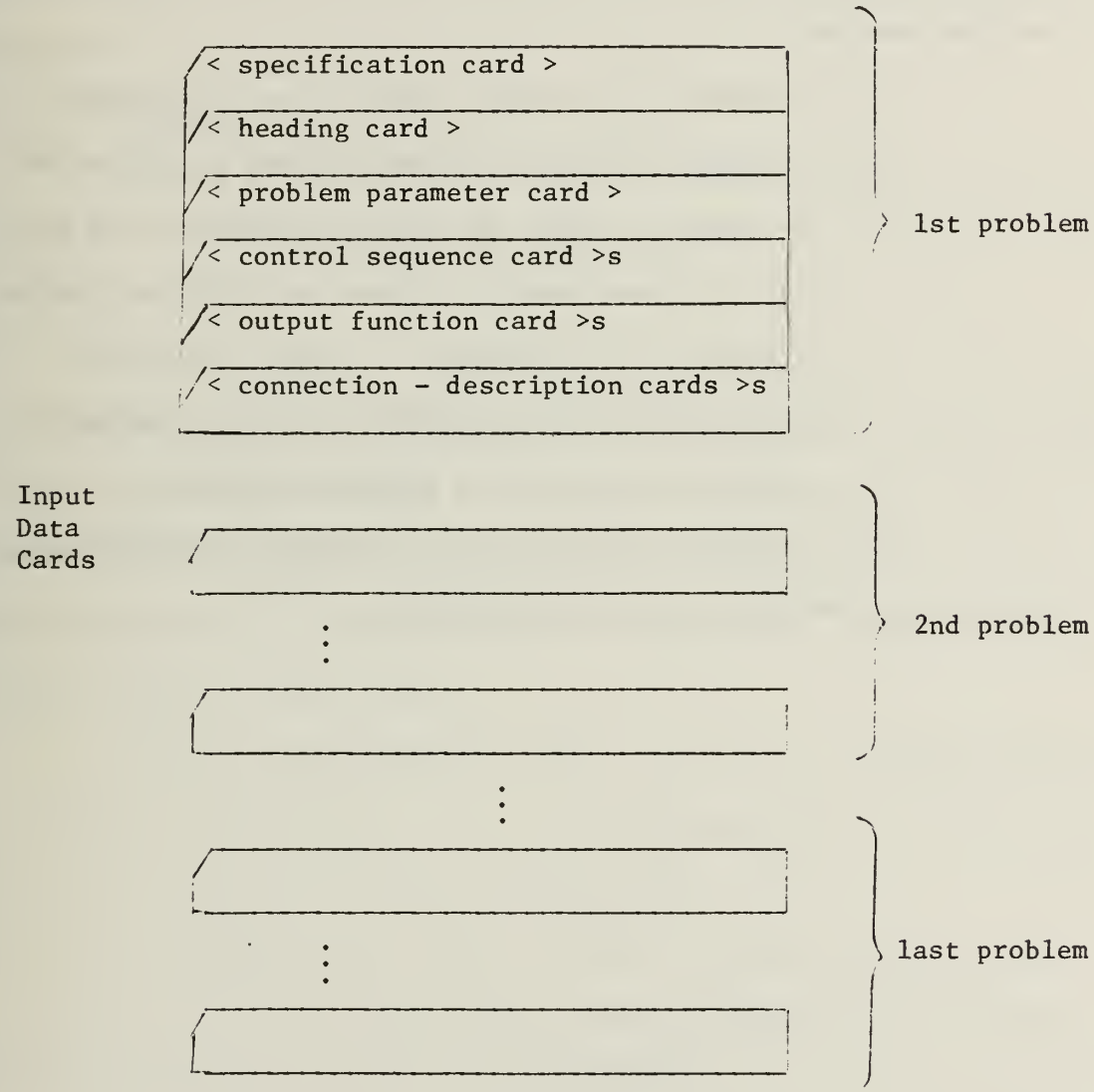
These 6 groups of cards, (i), (ii), (iii), (iv), (v) and (vi), in this order constitute the necessary description for a single problem. In order to solve more than one problem in one computer run, the user can arrange serially the descriptions for other problems. Fig. 2-2 shows the input card sequence for the execution of more than one problem using typical JCL statements for the IBM 360/75.

```
/*ID ...                    } JCL cards for problem identification
  :
  :
/*ID SETUP UNIT=2314,ID=DK0043

// EXEC  GOFORT.PROG=SYSTEM1,PARM=/OVLY,LIST,XCAL',REGION=390K

//GO.STEPLIB  DD DSN=USER.P1189.NETTRA,DISP=SHR

//GO.SYSIN  DD  *
```




```
/*
```

Fig. 2-2 Input card sequence for the execution of problems by the NETTRA system

3. RESTRICTIONS ON PROBLEM SIZE

In order to fit the program into a limited memory size (currently 400k bytes with IBM 360/75J), some restrictions on an acceptable size of a problem were made:

1. The number of external variables may not exceed 5.
2. The number m of output functions may not exceed 10*.
3. The number of gates, R, may not exceed 60-n in the case of uncomplemented external variables available (a blank or 'x' parameter). In the case of both complemented and uncomplemented external variables available, the limit is lowered to 60-2n.

All of these limitations are essentially imposed by the array sizes in the program as presently written.

*For initial network method TISN or TLEV, this is lowered to 4.

4. EXAMPLES OF INPUT DATA SETUP

The following examples will illustrate, for using the NETTRA system, various possible input data card setups complying with the directions given in Chapter 2.

Example 4-1 A one-bit full adder.

One-bit full adder is an important arithmetic unit in most digital computers. The output functions are SUM = $x_1 \oplus x_2 \oplus x_3$ and CARRY = $x_1x_2 \vee x_1x_3 \vee x_2x_3$.

In this example, we want to get results using the control sequence OPT1 but under different fan-in/fan-out restrictions (TFI=TFO=TFOX=TFOO=2 or 3 or 4). Assume that the goal of cost reduction is to reduce the number of gates primarily and the number of connections secondarily. Also assume that only uncomplemented external variables are available. The input data cards for this example is shown below.

(i) The < specification card > contains the following information:

cols. 1∿4	0,	no heading card for the following problems.
cols. 5∿8	0,	using the same functions as the first problem for the following problems.
cols. 9∿12	1,	using different parameters for each problem.
cols. 13∿16	0,	using the same control sequence for each problem.
cols. 17∿20	0,	no result will be punched.

cols. 21 24 600, for each problem the processing time is 600 seconds (10 minutes).

(ii) The < heading card > contains:

*** ONE-BIT FULL ADDER : S=EXOR(X1,X2,X3),C=X1X2 X1∨X3∨X2∨X3 ***

(iii) Three < parameter card >s are required for three sets of fan-in/fan-out restrictions. But the following parameters have the same values.

cols. 1∿4 3, the value of N.

cols. 5∿8 2, the value of m.

cols. 9∿12 100, the value of A.

cols. 13∿16 1, the value of B.

cols. 17∿20 X, uncomplemented external variables available only.

cols. 37∿40 "blank" use the default value for LREST.

cols. 41∿44 and

cols. 45∿48 "blank" no printout for intermediate steps.

cols. 49∿52 0, the problems are run for the first time.

cols. 53∿56 0, use the default for NGPMAX.

cols. 21∿24, 25∿28, 29∿32, 33∿36 are the parameters TFI, TFO, TFOX and TFOO. For our problem, they are set to the values 2's, 3's and 4's on each of the parameter cards.

(iv) The < control sequence card > in this example contains the keyword OPT1 only.

(v) The < output function card >s consist of two cards, one for the sum and another one for the carry. Their binary representatives are:

01101001 -- SUM

00010111 -- CARRY

Since we do not read in the network explicitly no < connection-description card >s are necessary.

The whole set of input cards for this problem is shown in Fig. 4-1.

col. 1 ↓

First Card →

```
  0   0   1   0   0 600
***  ONE-BIT FULL ADDER : S=EXOR(X1,X2,X3), C=X1X2∨X1X3∨X2X3  ***
  3   21000   1   X   2   2   2   2                0   0
OPT1
01101001
00010111
  3   21000   1   X   3   3   3   3                0   0
  3   21000   1   X   4   4   4   4                0   0
```

Fig. 4.1 Input data setup for Example 4-1.

Example 4-2 Parity functions

In this example, we want to find near-optimal networks for 3-variable, 4-variable and 5-variable odd parity functions. For these parity functions, same fan-in, fan-out restrictions but different control sequences are used. Assume that both complemented and uncomplemented external variables are available. The cost criterion is to reduce the number of gates primarily and the number of connections secondarily. The details of each card is explained below.

(i) The < specification card > has the following contents:

cols. 1∿4	1,	read a heading card for each problem.
cols. 5∿8	1,	read a function card for each problem.
cols. 9∿12	1,	read the parameter card for each problem.
cols. 13∿16	1,	read a control sequence card for each problem.

cols. 17∿20 0, no punch for the best results obtained.

cols. 21∿24 600, the maximum execution time is 10 minutes for each problem.

(ii) The contents for each < heading card > will be shown later.

(iii) The following fields of the < problem parameter card >s have the same information for the three problems.

cols. 5∿ 8 1, the value of M

cols. 9∿12 1000, the value of A

cols. 13∿16 1, the value of B

cols. 17∿20 Y, complemented and uncomplemented variables are available.

cols. 21∿24 4, the value of TFI

cols. 25∿28 4, the value of TFO.

cols. 29∿32 4, the value of TFOX.

cols. 33∿36 4, the value of TFOO.

cols. 37∿40 "blank", use the default value for LREST

cols. 41∿44 and

cols. 45∿48 "blank", no printout for intermediate steps.

cols. 49∿52 "blank", run the problem for the first time.

cols. 53∿56 "blank", use the default value for NEPMAX.

Only the first field has different values for N for different problems.

(iv) The following three different < control sequence card >s are for solving the 3-variable, 4-variable and 5-variable parity functions, respectively.

1. INTP BANB TDTP NTG1 # JEFF FLTP NTG1 # NOIT # STOP

2. INTP BANB TDTP NTG2 # JEFF FLTP NTG2 # NOIT # STOP

3. INTP BANB TDTP NTE1 # JEFF FLTP NTE1 # NOIT # STOP

(v) The < output function card > is different for each problem.

1. 01101001 $x_1 \oplus x_2 \oplus x_3$
2. 0110100110010110 $x_1 \oplus x_2 \oplus x_3 \oplus x_4$
3. 01101001100101101001011001101001 $x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5$

No < connection-description card >s are necessary.

Fig. 4.2 shows the input data setup for this example.

col. 1 ↓

First Card →

```
  1   1   1   1   0 600
***  THREE-VARIABLE PARITY FUNCTION  ***
  3   11000   1   Y   4   4   4   4
 INTP  BANB  TDTP  NTG1 #  JEFF  FLTP  NTG1 #  NOIT #  STOP
01101001
***  FOUR-VARIABLE PARITY FUNCTION  ***
  4   11000   1   Y   4   4   4   4
 INTP  BANB  TDTP  NTG2 #  JEFF  FLIP  NTG2 #  NOIT #  STOP
0110100110010110
***  FIVE-VARIABLE PARITY FUNCTION  ***
  5   11000   1   Y   4   4   4   4
 INTP  BANB  TDTP  NTE1 #  JEFF  FLIP  NTE1 #  NOIT #  STOP
01101001100101101001011001101001
```

Fig. 4.2 Input data setup for Example 4-2.

Example 4-3: Su-Nam's example functions

There are 4-variable 4-output incompletely specified functions used by Su and Nam in [3]. We want to solve this example by the NETTRA system and compare the result with that obtained in [3] which has 25 gates, 42 connections and 6 levels. The restrictions used in [3] are: FI = FO = FOX = 2 and FOO = 0 and both complemented and uncomplemented external variables are available. Again the cost reduction criterion is to reduce the number of gates primarily and the number of connections secondarily. Fig. 4-3 shows the input data cards.

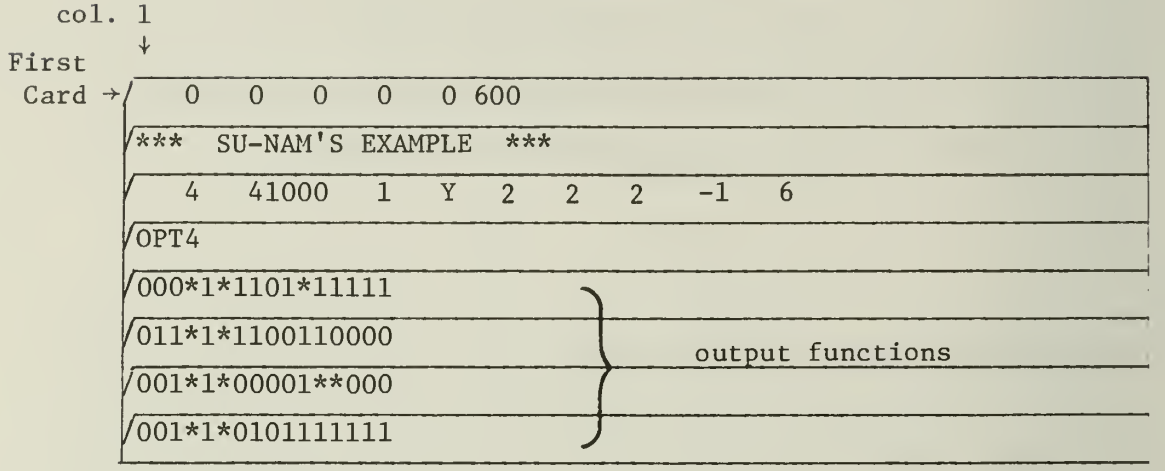


Fig. 4.3 Input data cards for Example 4-3.

Notice that the < specification card > for this example contains 0's in the first five fields. This is because only a single problem is involved in this run.

Example 4-4: In this example, we want to compare the effectiveness and the efficiency of the transduction procedures under no fan-in/fan-out restrictions. The 5-variable function $f(x_1,x_2,x_3,x_4,x_5) = \Sigma$ (0,4,5,6, 7,8,10,14,16,19,21,23,24,25,26,27,29,30) is used, and the control sequences composed of the initial network method UNIV and different transduction procedures are applied. Assume only uncomplemented external variables are available and the cost criterion is to reduce the number of connections primarily and the number of gates secondarily. The input data deck is shown in Fig. 4-4.

```
   col. 1
        ↓
First
 Card →  0   0   0   1   0 300
        *** FIVE-VARIABLE FUNCTION:HEX=8FA295F6  ***
          5   1   11000   X
         INTP  UNIV  TDTP  NPGM #  NOIT 1  STOP
        10001111101000101001010111110110
         INTP  UNIV  TDTP  NPGS #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTP1 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTP2 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTG1 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTG2 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTG3 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTG4 #  NOIT 1  STOP
         INTP  UNIV  TDTP  NTE1 #  NOIT 1  STOP
```

Fig. 4-4 Input data deck for Example 4-4.

In Fig. 4.4, the execution time for each problem is 300 seconds. No fan-in/van-out restrictions and level restriction are imposed.

Example 4-5: In this example we want to get initial networks by different initial network methods. Since the initial network method TLEV is for the level-restricted case, it is not included in the control sequence. The same output function used in Example 4-4 is selected. Assume only uncomplemented external variables are available.

The input data deck is shown in Fig. 4-5. On the < problem parameter card >, the values of A and B are 1000 and 1, respectively. The values of A and B are not useful since no transduction or transformation is involved. But they are specified for printing the network costs. Another thing which should be noticed is that NOIT still has to be specified although no TT-sequence is included.

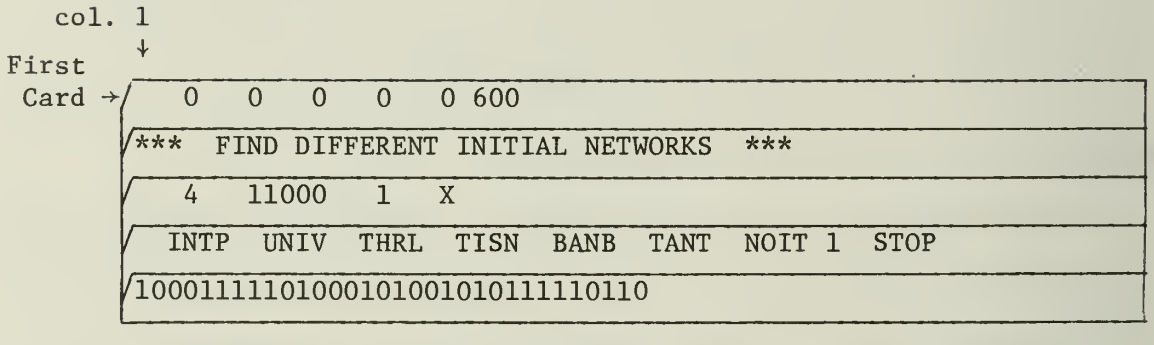


Fig. 4-5 Input data deck for example 4-5.

5. OUTPUT FORMATS

There are two types of output formats for the NETTRA system. One is for problems using the built-in control sequences OPT1, OPT2 and OPT3, and the other is for problems using the built-in control sequence OPT4 and the control sequences designed by the users themself.

Example 4-1 uses the control sequence OPT1 to find near-optimal networks for the one-bit full adder under different fan-in/fan-out restrictions. The printout for this example is shown in Fig. 5-1(a)∿(e).

On the first page of the print-out for the first problem, the problem number is shown first. Then the contents of the < heading card >, the number of external variables(n), the number of output functions(m), the cost coefficients(A and B), the status of external variables(whether uncomplemented and complemented external variables are available), the fan-in, fan-out and level restrictions the computation time specified for each problem in this run, the contents of the control sequence and the output functions(in row vector form) are printed one by one.

The cost, the computation time, and the network configuration for the best network are shown following the above printout. The network configuration is described in a way similar to the specification for the < connection-description card >s: describing the input connections for each gate in the network. The gate number, the gate level and the immediate predecessors of a gate are shown in the columns entitled with "GATE", "LEVEL" and "FED BY", respectively. The gates, which exist in the initial network or in the network just after the application of the transformation procedure JEFF but then become redundant after the applications of the

```
**********************
*  PROBLEM NO.    1  *
**********************
```

*** ONE-BIT FULL ADDER : S= EXOR(X1,X2,X3), C=X1X2 V X1X3 V X2X3 ***

NUMBER OF VARIABLES = 3
NUMBER OF FUNCTIONS = 2
COST COEFFICIENT A = 1000
B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 2
FAN-OUT = 2
FAN-OUT FOR EX. VARIABLES = 2
FAN-OUT FOR OUTPUT GATES = 2
NO. OF LEVELS = 100
TIME SPECIFIED FOR THE PROBLEM = 60000 CS

CONTROL SEQUENCE SPECIFIED BY THE USER

OPT1

FUNCTION NO. 1
01101001
FUNCTION NO. 2
00010111

Fig. 5-1(a)

```
THIS IS THE BEST RESULT FOUND BY CONTROL SEQUENCE OPTION  1

TOTAL TIME SPENT =     10742 CENTISECONDS
BEST COST =     12019


TRUTH TABLE
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL        FED BY
   1       / 1/        8 19
   2       / 1/       11
   3       / 5/        4  6
   4       / 6/       X1 X2
   5       / 7/       X2
   6       / 6/        5  7
   7       / 7/       X1
   8       / 3/        9 12
   9       / 4/        5
  10       / 1/
  11       / 2/        6  8
  12       / 4/       X3
  13       / 1/
  14       / 1/
  15       / 1/
  16       / 1/
  17       / 1/
  18       / 1/
  19       / 2/       X3  3
  20       / 1/
  21       / 1/
  22       / 1/
  23       / 1/
  24       / 1/
  25       / 1/
  26       / 1/
  27       / 1/
  28       / 1/
  29       / 1/
  30       / 1/
```

Fig. 5-1(b)

```
**********************
*  PROBLEM NO.    2  *
**********************
```

NUMBER OF VARIABLES = 3

NUMBER OF FUNCTIONS = 2

COST COEFFICIENT A = 1000

B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 3
FAN-OUT = 3
FAN-OUT FOR EX. VARIABLES = 3
FAN-OUT FOR OUTPUT GATES =
NO. OF LEVELS = 100

THIS IS THE BEST RESULT FOUND BY CONTROL SEQUENCE OPTION 1

TOTAL TIME SPENT = 1600 CENTISECONDS

BEST COST = 9018

TRUTH TABLE

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1
```

GATE ..	LEVEL	FED BY
1	/ 1/	3 5
2	/ 1/	6 7
3	/ 2/	X2 6
4	/ 1/	
5	/ 2/	6 7 8
6	/ 3/	X2 7 8
7	/ 4/	X1 X3
8	/ 4/	9 10
9	/ 5/	X3
10	/ 5/	X1
11	/ 1/	
12	/ 1/	
13	/ 1/	

Fig. 5-1(c)

```
**********************
*  PROBLEM NO.   3   *
**********************
```

NUMBER OF VARIABLES = 3

NUMBER OF FUNCTIONS = 2

COST COEFFICIENT A = 1000

B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 4
FAN-OUT = 4
FAN-OUT FOR EX. VARIABLES = 4
FAN-OUT FOR OUTPUT GATES = 4
NO. OF LEVELS = 100

THIS IS THE BEST RESULT FOUND BY CONTROL SEQUENCE OPTION 1

TOTAL TIME SPENT = 661 CENTISECONDS

BEST COST = 9018

TRUTH TABLE

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1
```

IE ..	LEVEL	FED BY
1	/ 1/	3 5
2	/ 1/	6 7
3	/ 2/	X2 6
4	/ 1/	
5	/ 2/	6 7 8
6	/ 3/	X2 7 8
7	/ 4/	X1 X3
8	/ 4/	9 10
9	/ 5/	X3
10	/ 5/	X1
11	/ 1/	
12	/ 1/	
13	/ 1/	

Fig. 5-1(d)

****
* PROBLEM NO. 4 *
****

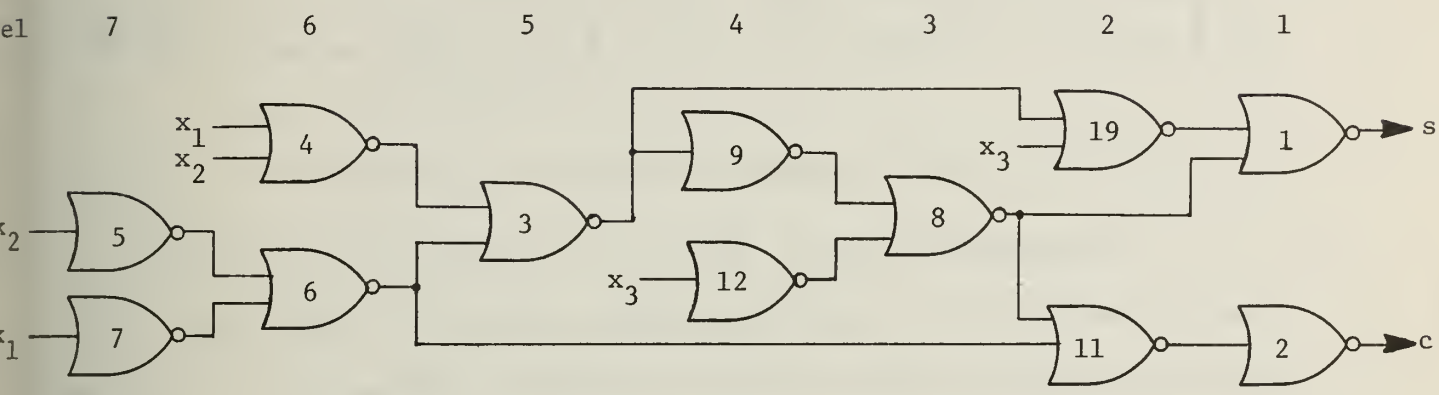
***** END OF FILE *****

****
* SUMMARY FOR THIS RUN *
****

COUN. NO.	N	M	A	B	FI	FO	FOX	FOO	LREST	BEST COST	NO. OF LEV.	TIME(CS)
1	3	2	1000	1	2	2	2	2	100	12019	7	10742
2	3	2	1000	1	3	3	3	3	100	9018	5	1500
3	3	2	1000	1	4	4	4	4	100	9018	5	661

Fig. 5-1(e)

transduction procedures, are assigned as the first-level gates and they have no immediate predecessors. The best network for the first problem is shown in Fig. 5-2.



ig. 5-2 The best network obtained for the first problem in Example 5-1. Each gate has a gate number corresponding to the network configuration.

For problems 2 and 3, only the information on the <problem parameter card > is shown, since the heading information, the control sequence and the output functions are the same as those of the first problem. The best networks for problems 2 and 3 are found identical, and the network configuration is shown in Fig. 5-3. It is interesting to notice that the networks obtained in Fig. 5-3 has the minimal number of gates under given restrictions. The optimality is proved by the integer programming logic design method based on branch-and-bound method.

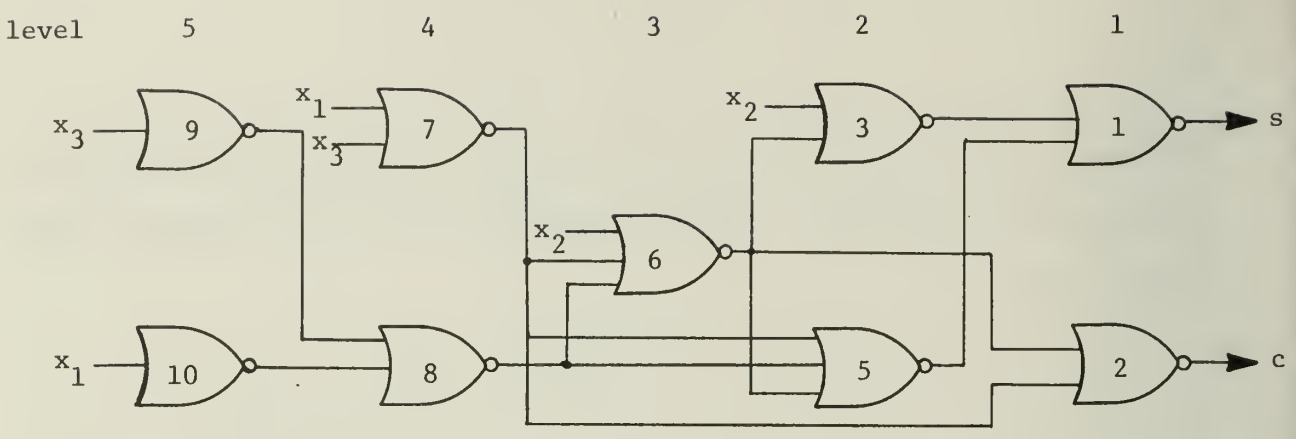


Fig. 5-3 The best network for problems $\geq$ in Example 4-1.

The problem number for the fourth problem is printed after the results for the first three problems have been printed. Since there is no more input data card, "END OF FILE" is printed to indicate that all problems have been processed. A table, shown in Fig. 5-1(e) is then printed to give a clear summary for this computer run. This summary contains the information about the problem size, fan-in, fan-out and level restrictions, the costs of the best networks, the numbers of levels of the networks and the computation time spent.

For the second type of printouts which correspond to the use of the built-in control sequence OPT4 or the control sequences designed by the users, the results of the intermediate steps will be printed as well as the printout just described for the first type printout. In other words, the costs, the computation times, the network configurations of the initial networks, the networks after applying the transformation and the

transductions will be printed. Fig. 5-4 (a)∿(z) shows the printout for the problem which uses the following control sequence:

INTP BANB THRL TDTP NPGM # NTG3 # JEFF FLIP NTG1 # NTG2 # NOIT # STOP

The output function is the 5-variable single-output function $f = \Sigma (0,4,5,6,7,8,10,14,16,19,21,23,24,25,26,27,29,30)$. In Fig 5-4(a), the information about the problem is shown. The initial network obtained by the branch-and-bound algorithm is printed in Fig. 5-4(b). In Fig. 5-4(b) and (c), the results after applying NPGM two iterations and then NTG3 two iterations are shown. Then the results after applying JEFF and the transduction procedures are shown one by one. In Fig. 5-4(0), the network with cost 12030 after applying the TT-sequence twice is obtained. Since the network cannot be simplified further, we go back to try the initial network method THRL and apply the same TT-sequence. Finally, in Fig. 5-4(y), a network with cost 13032 is obtained. The summary is shown in Fig. 5-4(z).

For the problems using the built-in control sequence OPT4, the output formats are the same as the previous example, i.e., the results of the intermediate steps are printed. Besides, the status (whether the network is fan-in, fan-out restricted and level-restricted or not) of the initial network, and the network after each application of the transduction is also shown. Fig. 5-5(a)∿(t) gives the results of the one-bit full adder under both fan-in/fan-out and level restrictions using the control sequence OPT4. The first problem, Fig. 5-5(a)∿(g), is under the restrictions TFI = TFO = TFOX = TFOO = 3 and LREST = 4. The initial network derived by TISLEV is fan-in/fan-out restricted and level-restricted. So after application of the transduction procedures, a feasible network with cost 12024 is obtained in Fig.

```
**********************
*  PROBLEM NO.    1  *
**********************
```

*** FIVE-VARIABLE FUNCTION HEX=8FA295F6 ***

NUMBER OF VARIABLES = 5
NUMBER OF FUNCTIONS = 1
COST COEFFICIENT A = 1000
B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 4
FAN-OUT = 4
FAN-OUT FOR EX. VARIABLES = 4
FAN-OUT FOR OUTPUT GATES = 4
NO. OF LEVELS = 100
TIME SPECIFIED FOR THE PROBLEM = 60000 CS

CONTROL SEQUENCE SPECIFIED BY THE USER

INTP BANB THRL TDTP NPGM # NTG3 # JEFF FLTP NTG1 # NTG2 # NOIT # STOP

FUNCTION NO. 1
10001111101000101001010111110110

```
THIS INITIAL NETWORK IS FOUND BY BRANCH AND BOUND ALGORITHM

INITIAL NETWORK  COST = 14050
TIME ELAPSED  =        59  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0


TE .. LEVEL       FED BY
 1      / 1/       2
 2      / 2/       3   4   5   6
 3      / 3/      X5   9  10  11
 4      / 3/      X4   7   8   9  11  12
 5      / 5/      X3   7   8  10
 6      / 5/      X2   7   8   9  10
 7      / 8/      X1  X3  14
 8      / 7/      X2  X3  X4
 9      / 6/      X2  X4  X5   8  13
10      / 6/      X2  X5   8  13
11      / 4/      X4  X5   5   6   8
12      / 4/      X1   6
13      / 7/      X1   7
14      / 9/      X4  X5


NETWORK DERIVED BY MINI2-NO FANIN/FANOUT LIMITS

IN   2  ITERATIONS   COST = 13038
TIME ELAPSED  =        17  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0
```

Fig. 5-4(b)

GATE ..	LEVEL	FED BY
1	/ 1/	2
2	/ 2/	3 4 5 6
3	/ 3/	X5 10 11
4	/ 3/	X4 8 11 12
5	/ 3/	X3 7 8 10
6	/ 3/	X2 7 8 10
7	/ 6/	X1 X3 14
8	/ 5/	X2 X3 X4
9	/ 1/	
10	/ 4/	X2 X5 8 13
11	/ 4/	X4 X5 8
12	/ 4/	X1
13	/ 5/	X1 7
14	/ 7/	X4 X5

NETWORK DERIVED BY GATE MERGING-NO FANIN/FANOUT LIMITS

IN 2 ITERATIONS COST = 12036
TIME ELAPSED = 40 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1

GATE ..	LEVEL	FED BY
1	/ 1/	2
2	/ 2/	3 4 5 6
3	/ 3/	X5 10 11
4	/ 3/	X4 8 11 12
5	/ 3/	X3 7 8 10
6	/ 3/	X2 7 8 10
7	/ 6/	X1 X3 11
8	/ 8/	X2 X3 X4
9	/ 1/	
10	/ 4/	X2 X5 8 13
11	/ 7/	X4 X5 8
12	/ 4/	X1
13	/ 5/	X1 7
14	/ 1/	

Fig. 5-4(c)

A NETWORK DERIVED BY JEFF COST = 13039
TIME ELAPSED = 9 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2
2	/ 2/	3 4 5 6
3	/ 3/	X5 10 11
4	/ 3/	X4 9 11 12
5	/ 3/	X3 7 8 10
6	/ 3/	X2 7 8 10
7	/ 6/	X1 X3 11
8	/ 8/	X2 X3 X4
9	/ 4/	X2 X3 X4
10	/ 4/	X2 X5 8 13
11	/ 7/	X4 X5 8
12	/ 4/	X1
13	/ 5/	X1 7
14	/ 1/	

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 13037
TIME ELAPSED = 30 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 9 11 12
5	/ 3/	X3 7 9
6	/ 3/	X2 7 9
7	/ 4/	X1 X3 11
8	/ 6/	X2 X3 X4
9	/ 4/	X2 X3 X4
10	/ 2/	X2 X5 8 13
11	/ 5/	X4 X5 8
12	/ 4/	X1
13	/ 3/	X1 7
14	/ 1/	

Fig. 5-4(d)

A NETWORK DERIVED BY 2-TH APPLICATION OF PRIFF

COST = 13036
TIME ELAPSED = 22 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 8 11 12
5	/ 3/	X3 7 8
6	/ 3/	X2 7 8
7	/ 4/	X1 X3 11
8	/ 6/	X2 X3 X4
9	/ 3/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 4/	X1
13	/ 3/	X1 7
14	/ 1/	

A NETWORK DERIVED BY 3-TH APPLICATION OF PRIFF

COST = 13033
TIME ELAPSED = 27 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

Fig. 5-4(e)

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 8 11 12
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 4/	X1
13	/ 3/	X1 8
14	/ 1/	

A NETWORK DERIVED BY 4-TH APPLICATION OF PRIFF

COST = 12033
TIME ELAPSED = 17 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 7 8 11
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X2 X3 X4
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 3/	X1 8
14	/ 1/	

Fig. 5-4(f)

A NETWORK DERIVED BY 5-TH APPLICATION OF PRIFF

COST = 12033
TIME ELAPSED = 27 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 C 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 9 11 13
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X2 X3 X4
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 4/	X1 8
14	/ 1/	

** NO REDUNDANCY FOUND **

A NETWORK DERIVED BY 1-TH APPLICATION OF PROCIV

COST = 12033
TIME ELAPSED = 132 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1

Fig. 5-4(g)

```
GATE .. LEVEL        FED BY

   1       / 1/       2 10
   2       / 2/       3  4   5   6
   3       / 3/      X5 11
   4       / 3/      X4  7   9 11
   5       / 3/      X3  7   9
   6       / 3/      X2  8
   7       / 4/      X1 11
   8       / 6/      X2 X3
   9       / 4/      X2 X3  X4
  10       / 2/      X2 X5   9 13
  11       / 5/      X4 X5   8
  12       / 1/
  13       / 3/      X1  8
  14       / 1/
```

** NO REDUNDANCY FOUND **

NETWORK DERIVED BY MINI2-NO FANIN/FANOUT LIMITS

IN 1 ITERATIONS COST = 12033
TIME ELAPSED = 5 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

```
GATE .. LEVEL        FED BY

   1       / 1/       2 10
   2       / 2/       3  4   5   6
   3       / 3/      X5 11
   4       / 3/      X4  7   9 11
   5       / 3/      X3  7   9
   6       / 3/      X2  8
   7       / 4/      X1 11
   8       / 6/      X2 X3
   9       / 4/      X2 X3  X4
  10       / 2/      X2 X5   9 13
  11       / 5/      X4 X5   8
  12       / 1/
  13       / 3/      X1  8
  14       / 1/
```

Fig. 5-4(h)

```
NETWORK DERIVED BY GATE MERGING-NO FANIN/FANOUT LIMITS

IN    1  ITERATICNS    COST = 12033
TIME ELAPSED  =        15  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 (


GATE .. LEVEL       FED BY
   1      / 1/       2 10
   2      / 2/       3  4   5   6
   3      / 3/      X5 11
   4      / 3/      X4  7   9 11
   5      / 3/      X3  7   9
   6      / 3/      X2  8
   7      / 4/      X1 11
   8      / 6/      X2 X3
   9      / 4/      X2 X3  X4
  10      / 2/      X2 X5   9 13
  11      / 5/      X4 X5   8
  12      / 1/
  13      / 3/      X1  8
  14      / 1/


A NETWORK DERIVED BY JEFF   COST = 12033
TIME ELAPSED  =        4  CENTISECONDS


IRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1
```

Fig. 5-4(i)

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 7 9 11
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X2 X3 X4
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 3/	X1 8
14	/ 1/	

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 12032
TIME ELAPSED = 25 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 8 11 13
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 4/	X1 8
14	/ 1/	

Fig. 5-4(j)

A NETWORK DERIVED BY 2-TH APPLICATION OF PRIFF

COST = 12032
TIME ELAPSED = 25 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 8 11 13
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 4/	X1 8
14	/ 1/	

** NO REDUNDANCY FOUND **

A NETWORK DERIVED BY 1-TH APPLICATION OF PROCIV

COST = 12032
TIME ELAPSED = 142 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

Fig. 5-4(k)

```
GATE .. LEVEL       FED BY

   1      / 1/       2 10
   2      / 2/       3  4   5  6
   3      / 3/      X5 11
   4      / 3/      X4  7   8 11
   5      / 3/      X3  7   9
   6      / 3/      X2  8
   7      / 4/      X1 11
   8      / 6/      X2 X3
   9      / 4/      X4 11
  10      / 2/      X2 X5   9 13
  11      / 5/      X4 X5   8
  12      / 1/
  13      / 3/      X1  8
  14      / 1/
```

** NO REDUNDANCY FOUND **

NETWORK DERIVED BY MINI2-NO FANIN/FANOUT LIMITS

IN 1 ITERATIONS COST = 12032
TIME ELAPSED = 5 CENTISECONDS

TRUTH TABLE

```
I = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0
```

```
GATE .. LEVEL       FED BY

   1      / 1/       2 10
   2      / 2/       3  4   5  6
   3      / 3/      X5 11
   4      / 3/      X4  7   8 11
   5      / 3/      X3  7   9
   6      / 3/      X2  8
   7      / 4/      X1 11
   8      / 6/      X2 X3
   9      / 4/      X4 11
  10      / 2/      X2 X5   9 13
  11      / 5/      X4 X5   8
  12      / 1/
  13      / 3/      X1  8
  14      / 1/
```

Fig. 5-4(ℓ)

```
NETWORK DERIVED BY GATE MERGING-NO FANIN/FANOUT LIMITS

IN    1  ITERATIONS   COST = 12032
TIME ELAPSED  =        17  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE .. LEVEL       FED BY
   1      / 1/       2 10
   2      / 2/       3  4   5  6
   3      / 3/      X5 11
   4      / 3/      X4  7   8 11
   5      / 3/      X3  7   9
   6      / 3/      X2  8
   7      / 4/      X1 11
   8      / 6/      Y2 X3
   9      / 4/      X4 11
  10      / 2/      X2 X5   9 13
  11      / 5/      X4 X5   8
  12      / 1/
  13      / 3/      X1  8
  14      / 1/


A NETWORK DERIVED BY JEFF   COST = 12032
TIME ELAPSED  =        2  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0
```

Fig. 5-4(m)

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 7 8 11
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 3/	X1 8
14	/ 1/	

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 12032
TIME ELAPSED = 32 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 7 8 11
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 3/	X1 8
14	/ 1/	

Fig. 5-4(n)

** NO REDUNDANCY FOUND **

A NETWORK DERIVED BY 1-TH APPLICATION OF PROCIV

COST = 12032
TIME ELAPSED = 135 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 (

GATE ..	LEVEL	FED BY
1	/ 1/	2 10
2	/ 2/	3 4 5 6
3	/ 3/	X5 11
4	/ 3/	X4 8 11 13
5	/ 3/	X3 7 9
6	/ 3/	X2 8
7	/ 4/	X1 11
8	/ 6/	X2 X3
9	/ 4/	X4 11
10	/ 2/	X2 X5 9 13
11	/ 5/	X4 X5 8
12	/ 1/	
13	/ 4/	X1 8
14	/ 1/	

** NO REDUNDANCY FOUND **

Fig. 5-4(o)

```
          ***  TRY ANOTHER INITIAL NETWORK   ***


          THIS INITIAL NETWORK IS FOUND BY THREE-LEVEL-NETWORK ALGORITHM

          INITIAL NETWORK  COST = 26103
          TIME ELAPSED  =        10  CENTISECONDS


  TRUTH TABLE
    1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0


 GATE .. LEVEL       FED BY
    1       / 1/       3   4   5   7   9 11 13 15 16 17 19 21 26
    2       / 3/      X1  X2  X3  X4  X5
    3       / 2/      X1  X2  X3  X4   2
    4       / 2/      X1  X2  X3  X5   2
    5       / 2/      X1  X2  X3   2
    6       / 3/      X1  X3  X4  X5
    7       / 2/      X1  X3  X4   6
    8       / 3/      X1  X3  X5
    9       / 2/      X1  X3   8
   10       / 3/      X1  X2  X4  X5
   11       / 2/      X1  X4  X5   6  10
   12       / 3/      X1  X2  X4
   13       / 2/      X1  X4   6  12
   14       / 3/      X2  X3  X4  X5
   15       / 2/      X2  X3  X4  14
   16       / 2/      X2  X3  X5   2  14
   17       / 2/      X2  X4  X5   2  10  14
   18       / 3/      X1  X2  X5
   19       / 2/      X2  X5  14  18
   20       / 3/      X3  X4  X5
   21       / 2/      X4  X5  10  20
   22       / 3/      X2
   23       / 3/      X3
   24       / 3/      X4
   25       / 3/      X5
   26       / 2/      22  23  24  25
```

Fig. 5-4(p)

NETWORK DERIVED BY MINI2-NO FANIN/FANOUT LIMITS

IN 2 ITERATIONS COST = 19047
TIME ELAPSED = 28 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1

GATE ..	LEVEL	FED BY
1	/ 1/	9 13 15 16 19 21 26
2	/ 1/	
3	/ 1/	
4	/ 1/	
5	/ 1/	
6	/ 3/	X3
7	/ 1/	
8	/ 3/	X5
9	/ 2/	X1 X3 8
10	/ 3/	X2
11	/ 1/	
12	/ 3/	X2
13	/ 2/	X1 X4 6 12
14	/ 3/	X3 X4 X5
15	/ 2/	X2 X3 X4 14
16	/ 2/	X2 X3 X5 14
17	/ 1/	
18	/ 3/	X1
19	/ 2/	X2 X5 14 18
20	/ 3/	X3
21	/ 2/	X4 X5 10 20
22	/ 3/	X2
23	/ 3/	X3
24	/ 3/	X4
25	/ 3/	X5
26	/ 2/	22 23 24 25

NETWORK DERIVED BY GATE MERGING-NO FANIN/FANOUT LIMITS

IN 2 ITERATIONS COST = 14042
TIME ELAPSED = 173 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1

Fig 5-4(q)

GATE ..	LEVEL	FED BY
1	/ 1/	9 13 15 16 19 21 26
2	/ 1/	
3	/ 1/	
4	/ 1/	
5	/ 1/	
6	/ 3/	X1 X5
7	/ 1/	
8	/ 1/	
9	/ 2/	X1 X3 6
10	/ 3/	X2
11	/ 1/	
12	/ 1/	
13	/ 2/	X1 6 10
14	/ 3/	X3 X4 X5
15	/ 2/	X2 X3 X4 14
16	/ 2/	X2 X3 X5 14
17	/ 1/	
18	/ 1/	
19	/ 2/	X2 X5 6 14
20	/ 1/	
21	/ 2/	X4 X5 10 14
22	/ 1/	
23	/ 3/	X3
24	/ 3/	X4
25	/ 3/	X5
26	/ 2/	10 23 24 25

A NETWORK DERIVED BY JEFF COST = 19046
TIME ELAPSED = 22 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	7 19 21 26
2	/ 7/	X3 X5
3	/ 6/	2
4	/ 3/	25
5	/ 3/	9 13 15 16
6	/ 5/	X1 X5
7	/ 2/	5
8	/ 1/	
9	/ 4/	X1 X3 6
10	/ 5/	X2
11	/ 1/	

Fig. 5-4(r)

```
12      / 1/
13      / 4/        X1   6  10
14      / 5/        X4   3
15      / 4/        X2  X3  X4  14
16      / 4/        X2   3  14
17      / 1/
18      / 1/
19      / 2/        X2   4   6  14
20      / 1/
21      / 2/        X4  X5  10  14
22      / 1/
23      / 3/        X3
24      / 3/        X4
25      / 4/        X5
26      / 2/        10  23  24  25
```

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 19046
TIME ELAPSED = 54 CENTISECONDS

TRUTH TABLE

```
1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 10
```

```
GATE .. LEVEL       FED BY
   1      / 1/       7  19  21  26
   2      / 6/       X3  X5
   3      / 5/       X1   2
   4      / 3/       25
   5      / 3/        9  13  15  16
   6      / 3/       X1
   7      / 2/        5
   8      / 1/
   9      / 4/       X1  X3  25
  10      / 5/       X2
  11      / 1/
  12      / 1/
  13      / 4/       X1  10  25
  14      / 5/       X4
  15      / 4/       X2  X3  X4   2
  16      / 4/       X2  X5   3  14
  17      / 1/
  18      / 1/
  19      / 2/       X2   4   6  23
  20      / 1/
  21      / 2/       X4   4  10  23
  22      / 1/
  23      / 3/       X3
  24      / 3/       X4
  25      / 5/       X5
  26      / 2/       10  23  24  25
```

Fig. 5-4(s)

```
** NO REDUNDANCY FOUND **


          A NETWORK DERIVED BY  1-TH APPLICATION OF PROCIV

          COST = 13032
          TIME ELAPSED  =      629  CENTISECONDS


TRUTH TABLE
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1


GATE .. LEVEL       FED BY
    1      / 1/      7 15 19 26
    2      / 5/     X5
    3      / 3/     X1 X2 23
    4      / 1/
    5      / 3/      9 16
    6      / 1/
    7      / 2/      3  5
    8      / 1/
    9      / 4/     X1  2
   10      / 3/     X2
   11      / 1/
   12      / 1/
   13      / 1/
   14      / 5/     X4
   15      / 2/     X2 X3 X4  2
   16      / 4/     X2 X5 14
   17      / 1/
   18      / 1/
   19      / 2/     X4 X5  3 23
   20      / 1/
   21      / 1/
   22      / 1/
   23      / 4/     X3
   24      / 1/
   25      / 1/
   26      / 2/      2 10 14 23
```

Fig. 5-4(t)

A NETWORK DERIVED BY 2-TH APPLICATION OF PROCIV

COST = 13032
TIME ELAPSED = 195 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 C 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	7 15 19 26
2	/ 5/	X5
3	/ 3/	X1 X2 23
4	/ 1/	
5	/ 3/	9 16
6	/ 1/	
7	/ 2/	3 5
8	/ 1/	
9	/ 4/	X1 2
10	/ 3/	X2
11	/ 1/	
12	/ 1/	
13	/ 1/	
14	/ 5/	X4
15	/ 2/	X2 X3 X4 2
16	/ 4/	X2 X5 14
17	/ 1/	
18	/ 1/	
19	/ 2/	X4 X5 3 23
20	/ 1/	
21	/ 1/	
22	/ 1/	
23	/ 4/	X3
24	/ 1/	
25	/ 1/	
26	/ 2/	2 10 14 23

** NO REDUNDANCY FOUND **

Fig. 5-4(u)

```
        NETWORK DERIVED BY MINI2-NO FANIN/FANOUT LIMITS

        IN    1  ITERATIONS    COST = 13032
        TIME ELAPSED  =          7  CENTISECONDS


TRUTH TABLE
  1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0


ATE .. LEVEL       FED BY
  1      / 1/       7 15 19 26
  2      / 5/      X5
  3      / 3/      X1 X2 23
  4      / 1/
  5      / 3/       9 16
  6      / 1/
  7      / 2/       3  5
  8      / 1/
  9      / 4/      X1  2
 10      / 3/      X2
 11      / 1/
 12      / 1/
 13      / 1/
 14      / 5/      X4
 15      / 2/      X2 X3 X4  2
 16      / 4/      X2 X5 14
 17      / 1/
 18      / 1/
 19      / 2/      X4 X5  3 23
 20      / 1/
 21      / 1/
 22      / 1/
 23      / 4/      X3
 24      / 1/
 25      / 1/
 26      / 2/       2 10 14 23
```

```
        NETWORK DERIVED BY GATE MERGING-NO FANIN/FANOUT LIMITS

        IN    1  ITERATIONS    COST = 13032
        TIME ELAPSED  =         15  CENTISECONDS


TRUTH TABLE
  1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0
```

Fig. 5-4(v)

GATE ..	LEVEL	FED BY
1	/ 1/	7 15 19 26
2	/ 5/	X5
3	/ 3/	X1 X2 23
4	/ 1/	
5	/ 3/	9 16
6	/ 1/	
7	/ 2/	3 5
8	/ 1/	
9	/ 4/	X1 2
10	/ 3/	X2
11	/ 1/	
12	/ 1/	
13	/ 1/	
14	/ 5/	X4
15	/ 2/	X2 X3 X4 2
16	/ 4/	X2 X5 14
17	/ 1/	
18	/ 1/	
19	/ 2/	X4 X5 3 23
20	/ 1/	
21	/ 1/	
22	/ 1/	
23	/ 4/	X3
24	/ 1/	
25	/ 1/	
26	/ 2/	2 10 14 23

A NETWORK DERIVED BY JEFF COST = 13032
TIME ELAPSED = 7 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	7 15 19 26
2	/ 5/	X5
3	/ 3/	X1 X2 23
4	/ 1/	
5	/ 3/	9 16
6	/ 1/	

Fig. 5-4(w)

```
  7      / 2/        3  5
  8      / 1/
  9      / 4/       X1  2
 10      / 3/       X2
 11      / 1/
 12      / 1/
 13      / 1/
 14      / 5/       X4
 15      / 2/       X2 X3 X4  2
 16      / 4/       X2 X5 14
 17      / 1/
 18      / 1/
 19      / 2/       X4 X5  3 23
 20      / 1/
 21      / 1/
 22      / 1/
 23      / 4/       X3
 24      / 1/
 25      / 1/
 26      / 2/        2 10 14 23
```

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 13032
TIME ELAPSED = 23 CENTISECONDS

TRUTH TABLE

```
 1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0
```

```
GATE .. LEVEL       FED BY
  1      / 1/        7 15 19 26
  2      / 5/       X5
  3      / 3/       X1 X2 23
  4      / 1/
  5      / 3/        9 16
  6      / 1/
  7      / 2/        3  5
  8      / 1/
  9      / 4/       X1  2
 10      / 3/       X2
 11      / 1/
 12      / 1/
 13      / 1/
 14      / 5/       X4
 15      / 2/       X2 X3 X4  2
 16      / 4/       X2 X5 14
 17      / 1/
 18      / 1/
 19      / 2/       X4 X5  3 23
 20      / 1/
 21      / 1/
 22      / 1/
 23      / 4/       X3
 24      / 1/
 25      / 1/
 26      / 2/        2 10 14 23
```

Fig. 5-4(x)

** NO REDUNDANCY FOUND **

A NETWORK DERIVED BY 1-TH APPLICATION OF PROCIV

COST = 13032
TIME ELAPSED = 190 CENTISECONDS

TRUTH TABLE

1 = 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0

GATE ..	LEVEL	FED BY
1	/ 1/	7 15 19 26
2	/ 5/	X5
3	/ 3/	X1 X2 23
4	/ 1/	
5	/ 3/	9 16
6	/ 1/	
7	/ 2/	3 5
8	/ 1/	
9	/ 4/	X1 2
10	/ 3/	X2
11	/ 1/	
12	/ 1/	
13	/ 1/	
14	/ 5/	X4
15	/ 2/	X2 X3 X4 2
16	/ 4/	X2 X5 14
17	/ 1/	
18	/ 1/	
19	/ 2/	X4 X5 3 23
20	/ 1/	
21	/ 1/	
22	/ 1/	
23	/ 4/	X3
24	/ 1/	
25	/ 1/	
26	/ 2/	2 10 14 23

** NO REDUNDANCY FOUND **

Fig. 5-4(y)

```
**********************
*  PROBLEM NO.    2  *
**********************
```

***** END OF FILE *****

```
****************************
*   SUMMARY FOR THIS RUN   *
****************************
```

COUN. NO.	N	M	A	P	FI	FO	FOX	FOO	LREST	BEST COST	NO. OF LEV.	TIME(CS)
1	5	1	1000	1	4	4	4	4	100	12032	6	787
2	5	1	1000	1	4	4	4	4	100	13032	5	1353

Fig. 5-4(z)

5-5(g). The second problem is under the restrictions TFI = TFO = TFOX = TFOO = 2 and LREST = 6. The initial network shown in Fig. 5-5(h) is level-restricted but not fan-out restricted: the fan-outs of external variable x_1 and x_2 exceed the limit. After application of the transduction procedures, the fan-out of external variable x_2 is reduced to 2 but the fan-out problem of external variable x_1 still cannot be solved. Therefore, another initial network with 7 levels is obtained in Fig. 5-5(n),(o). This initial network is fan-in/van-out restricted, although not level-restricted. The transduction procedures are applied to try to reduce the number of levels of this network. The final results is shown in Fig. 5-5(s) and (t), where the network cost is reduced, but the network is still not level-restricted. This means that no feasible network can be obtained by the control sequence OPT4 under the given restrictions. In the summary table, six asterisks are printed for problem 2 under the column entitled with BEST COST to mean that no feasible solution can be obtained.

```
**********************
*  PROBLEM NO.    1  *
**********************
```

*** ONE-BIT FULL ADDER : S=EXOR(X1,X2,X3), C=X1X2 V X1X3 V X2X3 ***

NUMBER OF VARIABLES = 3
NUMBER OF FUNCTIONS = 2
COST COEFFICIENT A = 1000
B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 3
FAN-OUT = 3
FAN-OUT FOR EX. VARIABLES = 3
FAN-OUT FOR OUTPUT GATES = 3
NO. OF LEVELS = 4
TIME SPECIFIED FOR THE PROBLEM = 60000 CS

CONTROL SEQUENCE SPECIFIED BY THE USER

OPT4

FUNCTION NO. 1
01101001
FUNCTION NO. 2
00010111

Fig. 5-5(a)

```
          THIS INITIAL NETWORK IS FOUND BY TISLEV METHOD

          INITIAL NETWORK  COST = 15029
          TIME ELAPSED  =        57  CENTISECONDS


  TRUTH TABLE

   1 = 0 1 1 0 1 0 0 1
   2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY

    1       / 1/       6  7  8
    2       / 1/      10 12 13
    3       / 4/      X1
    4       / 4/      X2
    5       / 4/      X3
    6       / 2/       3  9 10
    7       / 2/       4  5 14
    8       / 2/      14 15 16
    9       / 3/       4  5
   10       / 3/      X2 X3
   11       / 1/
   12       / 2/      14 16
   13       / 2/      X1 15
   14       / 3/       3
   15       / 3/       4
   16       / 3/       5

 ** LEVEL RESTRICTED **


          A NETWORK DERIVED BY  1-TH APPLICATION OF SUBSTI & MINI2

          COST = 15029
          TIME ELAPSED  =        12  CENTISECONDS


  TRUTH TABLE

   1 = 0 1 1 0 1 0 0 1
   2 = 0 0 0 1 0 1 1 1
```

Fig. 5-5(b)

```
GATE .. LEVEL        FED BY

   1       / 1/        6   7   8
   2       / 1/       10  12  13
   3       / 4/       X1
   4       / 4/       X2
   5       / 4/       X3
   6       / 2/        3   9  10
   7       / 2/        4   5  14
   8       / 2/       14  15  16
   9       / 3/        4   5
  10       / 3/       X2  X3
  11       / 1/
  12       / 2/       14  16
  13       / 2/       X1  15
  14       / 3/        3
  15       / 3/        4
  16       / 3/        5

 **   LEVEL RESTRICTED   **
```

A NETWORK DERIVED BY 1-TH APPLICATION OF GTMERG

COST = 13024
TIME ELAPSED = 32 CENTISECONDS

TRUTH TABLE

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1
```

```
GATE .. LEVEL        FED BY

   1       / 1/        6   7   8
   2       / 1/       10  12
   3       / 4/       X1
   4       / 4/       X2
   5       / 4/       X3
   6       / 2/        3   9  10
   7       / 2/        4   5  14
   8       / 2/       14  15
   9       / 3/        4   5
  10       / 4/       X2  X3
  11       / 1/
  12       / 2/        9  14
  13       / 1/
  14       / 3/        3
  15       / 3/       10
  16       / 1/

 **   LEVEL RESTRICTED   **
```

Fig. 5-5(c)

A NETWORK DERIVED BY 2-TH APPLICATION OF GTMERG

COST = 13024
TIME ELAPSED = 5 CENTISECONDS

TRUTH TABLE

```
1 = 0 1 1 0 1 0 0 1
2 = 0 0 0 1 0 1 1 1
```

GATE ..	LEVEL	FED BY
1	/ 1/	6 7 8
2	/ 1/	10 12
3	/ 4/	X1
4	/ 4/	X2
5	/ 4/	X3
6	/ 2/	3 9 10
7	/ 2/	4 5 14
8	/ 2/	14 15
9	/ 3/	4 5
10	/ 4/	X2 X3
11	/ 1/	
12	/ 2/	9 14
13	/ 1/	
14	/ 3/	3
15	/ 3/	10
16	/ 1/	

** LEVEL RESTRICTED **

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 12024
TIME ELAPSED = 15 CENTISECONDS

TRUTH TABLE

```
1 = 0 1 1 0 1 0 0 1
2 = 0 0 0 1 0 1 1 1
```

Fig. 5-5(d)

```
GATE .. LEVEL       FED BY
   1      / 1/       6   7   8
   2      / 1/      10  12
   3      / 4/      X1
   4      / 4/      X2
   5      / 4/      X3
   6      / 2/       3   9  10
   7      / 3/      X1   4   5
   8      / 2/      X1  X2  X3
   9      / 3/       4   5
  10      / 3/      X2  X3
  11      / 1/
  12      / 2/       7  14
  13      / 1/
  14      / 3/       3
  15      / 1/
  16      / 1/

 **   LEVEL RESTRICTED   **
```

A NETWORK DERIVED BY 2-TH APPLICATION OF PRIFF

COST = 12024
TIME ELAPSED = 22 CENTISECONDS

TRUTH TABLE

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1
```

```
GATE .. LEVEL       FED BY
   1      / 1/       6   7   8
   2      / 1/      10  12
   3      / 4/      X1
   4      / 4/      X2
   5      / 4/      X3
   6      / 2/       3   9  10
   7      / 2/       4   5  14
   8      / 2/      X2  X3  14
   9      / 3/       4   5
  10      / 3/      X2  X3
  11      / 1/
  12      / 2/      X1   9
  13      / 1/
  14      / 3/       3
  15      / 1/
  16      / 1/

 **   LEVEL RESTRICTED   **
```

Fig. 5-5(e)

```
          A NETWORK DERIVED BY  1-TH APPLICATION OF PROCIV

          COST = 12024
          TIME ELAPSED  =       82  CENTISECONDS


  TRUTH TABLE

   1 = 0 1 1 0 1 0 0 1
   2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY

    1      / 1/      6  7  8
    2      / 1/     10 12
    3      / 4/     X1
    4      / 4/     X2
    5      / 4/     X3
    6      / 2/      3  9 10
    7      / 3/     X1  4  5
    8      / 2/     Y1 X2 X3
    9      / 3/      4  5
   10      / 3/     X2 X3
   11      / 1/
   12      / 2/      7 14
   13      / 1/
   14      / 3/      3
   15      / 1/
   16      / 1/

 **  LEVEL RESTRICTED  **
```

```
          A NETWORK DERIVED BY  1-TH APPLICATION OF PROCCE

          COST = 12024
          TIME ELAPSED  =       85  CENTISECONDS
```

Fig. 5-5(f)

```
TRUTH TABLE
  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7  8
   2      / 1/      10 12
   3      / 4/      X1
   4      / 4/      X2
   5      / 4/      X3
   6      / 2/       3  9 10
   7      / 3/      X1  4  5
   8      / 2/      X1 X2 X3
   9      / 3/       4  5
  10      / 3/      X2 X3
  11      / 1/
  12      / 2/       7 14
  13      / 1/
  14      / 3/       3
  15      / 1/
  16      / 1/
 **   LEVEL RESTRICTED  **


** THIS IS A FEASIBLE SOLUTION **
```

```
**********************
*  PROBLEM NO.    2  *
**********************

NUMBER OF VARIABLES =    3
NUMBER OF FUNCTIONS =    2
COST COEFFICIENT A  = 1000
                  B =    1
```

Fig. 5-5(g)

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 2
FAN-OUT = 2
FAN-OUT FOR EX. VARIABLES =
FAN-OUT FOR OUTPUT GATES =
NO. OF LEVELS = 6

THIS INITIAL NETWORK IS FOUND BY TISLEV METHOD

INITIAL NETWORK COST = 18023
TIME ELAPSED = 117 CENTISECONDS

TRUTH TABLE

1 = 0 1 1 0 1 0 0 1
2 = 0 0 0 1 0 1 1 1

GATE ..	LEVEL	FED BY
1	/ 1/	6 7
2	/ 1/	8
3	/ 5/	X1
4	/ 1/	
5	/ 6/	X3
6	/ 2/	8 9
7	/ 2/	10 11
8	/ 3/	9 12
9	/ 4/	11 22
10	/ 3/	14 15
11	/ 5/	X2 22
12	/ 4/	5 17
13	/ 1/	
14	/ 4/	20 21
15	/ 4/	3 21
16	/ 1/	
17	/ 5/	X1 X2
18	/ 1/	
19	/ 6/	X2
20	/ 5/	19
21	/ 5/	5
22	/ 6/	X1

FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 = 3

FAN-OUT PROBLEM IN EXTERNAL VARIABLE X2 = 3

** LEVEL RESTRICTED **

Fig. 5-5(h)

```
A NETWORK DERIVED BY  1-TH APPLICATION OF SUBSTI & MINI2

COST = 17028
TIME ELAPSED  =        32  CENTISECONDS


 TRUTH TABLE
  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 1/       8
   3      / 5/      X1
   4      / 1/
   5      / 6/      Y3
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       9 12
   9      / 4/      11 22
  10      / 3/      14 15
  11      / 5/      X2 22
  12      / 4/       5 17
  13      / 1/
  14      / 4/      20 21
  15      / 4/       3 21
  16      / 1/
  17      / 6/      X1 X2
  18      / 1/
  19      / 1/
  20      / 5/      17
  21      / 5/       5
  22      / 6/      X1
 FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
 **   IEVEL RESTRICTED  **
```

Fig. 5-5(i)

```
          A NETWORK DERIVED BY  2-TH APPLICATION OF SUBSTI & MINI

          COST = 17028
          TIME ELAPSED  =        14  CENTISECONDS


  TRUTH TABLE
    1 = 0 1 1 0 1 0 0 1
    2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
    1      / 1/       6  7
    2      / 1/       8
    3      / 5/      X1
    4      / 1/
    5      / 6/      X3
    6      / 2/       8  9
    7      / 2/      10 11
    8      / 3/       9 12
    9      / 4/      11 22
   10      / 3/      14 15
   11      / 5/      X2 22
   12      / 4/       5 17
   13      / 1/
   14      / 4/      20 21
   15      / 4/       3 21
   16      / 1/
   17      / 6/      X1 X2
   18      / 1/
   19      / 1/
   20      / 5/      17
   21      / 5/       5
   22      / 6/      X1
 FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
 **   LEVEL RESTRICTED  **



          A NETWORK DERIVED BY  1-TH APPLICATION OF GIMERG

          COST = 17028
          TIME ELAPSED  =        27  CENTISECONDS


  TRUTH TABLE
    1 = 0 1 1 0 1 0 0 1
    2 = 0 0 0 1 0 1 1 1
```

Fig. 5-5(j)

```
GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 1/       8
   3      / 5/      X1
   4      / 1/
   5      / 6/      X3
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       9 12
   9      / 4/      11 22
  10      / 3/      14 15
  11      / 5/      X2 22
  12      / 4/       5 17
  13      / 1/
  14      / 4/      20 21
  15      / 4/       3 21
  16      / 1/
  17      / 6/      X1 X2
  18      / 1/
  19      / 1/
  20      / 5/      17
  21      / 5/       5
  22      / 6/      X1
 FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
 **   LEVEL RESTRICTED  **


          A NETWORK DERIVED BY  1-TH APPLICATION OF PRIFF

          COST = 17028
          TIME ELAPSED  =       24  CENTISECONDS


  TRUTH TABLE

   1 = 0 1 1 0 1 0 0 1
   2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 1/       8
   3      / 5/      X1
   4      / 1/
   5      / 6/      X3
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       9 12
```

Fig. 5-5(k)

```
  9       / 4/          5 11
 10       / 3/         14 15
 11       / 5/         X2 22
 12       / 4/          5 17
 13       / 1/
 14       / 4/         X3 20
 15       / 4/         21 22
 16       / 1/
 17       / 6/         X1 X2
 18       / 1/
 19       / 1/
 20       / 5/         17
 21       / 5/          5
 22       / 6/         X1

FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
**  LEVEL RESTRICTED  **



          A NETWORK DERIVED BY  1-TH APPLICATION OF PROCIV

          COST = 16027
          TIME ELAPSED  =      187  CENTISECONDS


 TRUTH TABLE
  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
  1       / 1/          6  7
  2       / 1/          8
  3       / 5/         X1
  4       / 1/
  5       / 6/         X3
  6       / 2/          8  9
  7       / 2/         10 11
  8       / 3/          9 12
  9       / 4/         11 22
 10       / 3/         14 15
 11       / 5/         X2 22
 12       / 5/          5 17
 13       / 1/
 14       / 4/         X3 20
 15       / 4/          3 12
 16       / 1/
 17       / 6/         X1 X2
 18       / 1/
 19       / 1/
 20       / 5/         17
 21       / 1/
 22       / 6/         X1

FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
**  LEVEL RESTRICTED  **
```

Fig. 5-5(ℓ)

```
          A NETWORK DERIVED BY  2-TH APPLICATION OF PROCIV

          COST = 16027
          TIME ELAPSED  =      142  CENTISECONDS


  TRUTH TABLE
   1 = 0 1 1 0 1 0 0 1
   2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY

    1      / 1/       6  7
    2      / 1/       8
    3      / 5/      X1
    4      / 1/
    5      / 6/      X3
    6      / 2/       8  9
    7      / 2/      10 11
    8      / 3/       9 12
    9      / 4/       3 11
   10      / 3/      14 15
   11      / 5/      X2 22
   12      / 5/       5 17
   13      / 1/
   14      / 4/      X3 20
   15      / 4/       3 12
   16      / 1/
   17      / 6/      X1 X2
   18      / 1/
   19      / 1/
   20      / 5/      17
   21      / 1/
   22      / 6/      X1

 FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3
 **   LEVEL RESTRICTED   **
```

Fig. 5-5(m)

```
A NETWORK DERIVED BY  1-TH APPLICATION OF PROCCE

COST = 16027
TIME ELAPSED  =      197  CENTISECONDS


 TRUTH TABLE

  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY

   1      / 1/       6  7
   2      / 1/       8
   3      / 5/      X1
   4      / 1/
   5      / 6/      X3
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       9 12
   9      / 4/       3 11
  10      / 3/      14 15
  11      / 5/      X2 22
  12      / 5/       5 17
  13      / 1/
  14      / 4/      X3 20
  15      / 4/       3 12
  16      / 1/
  17      / 6/      X1 X2
  18      / 1/
  19      / 1/
  20      / 5/      17
  21      / 1/
  22      / 6/      X1

 FAN-OUT PROBLEM IN EXTERNAL VARIABLE X1 =    3

 **  LEVEL RESTRICTED  **



 ***  TRY ANOTHER INITIAL NETWORK   ***

INITIAL NETWORK  COST = 19030
TIME ELAPSED  =       75  CENTISECONDS
```

Fig. 5-5(n)

```
TRUTH TABLE
  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 4/      16 17
   3      / 7/      X1
   4      / 7/      X2
   5      / 1/
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       2
   9      / 6/       4 22
  10      / 3/      14 15
  11      / 3/      19 22
  12      / 1/
  13      / 1/
  14      / 4/      19 21
  15      / 4/       3 21
  16      / 5/      X3  9
  17      / 5/      X2 23
  18      / 1/
  19      / 5/       4
  20      / 6/      X3
  21      / 5/      20
  22      / 7/      X1
  23      / 6/       3
** NOT LEVEL RESTRICTED **


          A NETWORK DERIVED BY  1-TH APPLICATION OF SUBSTI & MINI2

          COST = 19030
          TIME ELAPSED  =       19  CENTISECONDS


TRUTH TABLE
  1 = 0 1 1 0 1 0 0 1
  2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 4/      16 17
   3      / 7/      X1
   4      / 7/      X2
```

Fig. 5-5(o)

```
  5       / 1/
  6       / 2/          8   9
  7       / 2/         10  11
  8       / 3/          2
  9       / 6/          4  22
 10       / 3/         14  15
 11       / 3/         19  22
 12       / 1/
 13       / 1/
 14       / 4/         19  21
 15       / 4/          3  21
 16       / 5/         X3   9
 17       / 5/         X2  23
 18       / 1/
 19       / 5/          4
 20       / 6/         X3
 21       / 5/         20
 22       / 7/         X1
 23       / 6/          3
**   NOT LEVEL RESTRICTED  **
```

A NETWOPK DEPIVED BY 1-TH APPLICATION OF GTMERG

COST = 19030
TIME ELAPSED = 34 CENTISECONDS

TRUTH TABLE

```
1 = 0 1 1 0 1 0 0 1
2 = 0 0 0 1 0 1 1 1
```

```
GATE .. LEVEL       FED BY
  1       / 1/          6   7
  2       / 4/         16  17
  3       / 7/         X1
  4       / 7/         X2
  5       / 1/
  6       / 2/          8   9
  7       / 2/         10  11
  8       / 3/          2
  9       / 6/          4  22
 10       / 3/         14  15
 11       / 3/         19  22
 12       / 1/
 13       / 1/
 14       / 4/         19  21
 15       / 4/          3  21
 16       / 5/         X3   9
 17       / 5/         X2  23
 18       / 1/
```

Fig. 5-5(p)

```
19      / 5/        4
20      / 6/       X3
21      / 5/       20
22      / 7/       X1
23      / 6/        3

**   NOT LEVEL RESTRICTED   **
```

A NETWORK DERIVED BY 1-TH APPLICATION OF PRIFF

COST = 19030
TIME ELAPSED = 22 CENTISECONDS

TRUTH TABLE

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1
```

```
GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 4/      16 17
   3      / 7/      X1
   4      / 7/      X2
   5      / 1/
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       2
   9      / 6/       3  4
  10      / 3/      14 15
  11      / 3/       2 17
  12      / 1/
  13      / 1/
  14      / 4/      19 21
  15      / 4/      21 22
  16      / 5/      X3  9
  17      / 5/      19 23
  18      / 1/
  19      / 6/       4
  20      / 6/      X3
  21      / 5/      20
  22      / 7/      Y1
  23      / 6/      22

**   NOT LEVEL RESTRICTED   **
```

Fig. 5-5(q)

A NETWORK DERIVED BY 1-TH APPLICATION OF PROCIV

COST = 13021
TIME ELAPSED = 238 CENTISECONDS

TRUTH TABLE

```
1 = 0 1 1 0 1 0 0 1
2 = 0 0 0 1 0 1 1 1
```

GATE ..	LEVEL	FED BY
1	/ 1/	6 7
2	/ 4/	16 17
3	/ 1/	
4	/ 7/	X2
5	/ 1/	
6	/ 2/	8 9
7	/ 2/	10 11
8	/ 3/	2
9	/ 6/	4 22
10	/ 3/	20
11	/ 3/	2 17
12	/ 1/	
13	/ 1/	
14	/ 1/	
15	/ 1/	
16	/ 5/	X3 9
17	/ 5/	X1 X2
18	/ 1/	
19	/ 1/	
20	/ 4/	X3
21	/ 1/	
22	/ 7/	X1
23	/ 1/	

** NOT LEVEL RESTRICTED **

A NETWORK DERIVED BY 2-TH APPLICATION OF PROCIV

COST = 13021
TIME ELAPSED = 120 CENTISECONDS

TRUTH TABLE

Fig. 5-5(r)

```
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 4/      16 17
   3      / 1/
   4      / 7/      X2
   5      / 1/
   6      / 2/       8  9
   7      / 2/      10 11
   8      / 3/       2
   9      / 6/       4 22
  10      / 3/      20
  11      / 3/       2 17
  12      / 1/
  13      / 1/
  14      / 1/
  15      / 1/
  16      / 5/      X3  9
  17      / 5/      X1 X2
  18      / 1/
  19      / 1/
  20      / 4/      X3
  21      / 1/
  22      / 7/      X1
  23      / 1/
 **   NOT LEVEL RESTRICTED  **


          A NETWORK DERIVED BY  1-TH APPLICATION OF PROCCE

          COST = 13021
          TIME ELAPSED  =      115  CENTISECONDS


 TRUTH TABLE
 1 = 0 1 1 0 1 0 0 1
 2 = 0 0 0 1 0 1 1 1


GATE .. LEVEL       FED BY
   1      / 1/       6  7
   2      / 4/      16 17
   3      / 1/
   4      / 7/      X2
   5      / 1/
   6      / 2/       8  9
```

Fig. 5-5(s)

```
 7      / 2/        10 11
 8      / 3/         2
 9      / 6/         4 22
10      / 3/        20
11      / 3/         2 17
12      / 1/
13      / 1/
14      / 1/
15      / 1/
16      / 5/        X3  9
17      / 5/        X1 X2
18      / 1/
19      / 1/
20      / 4/        X3
21      / 1/
22      / 7/        X1
23      / 1/
```

** NOT LEVEL RESTRICTED **

** WE CANNOT FIND A FEASIBLE SOLUTION **

****
* PROBLEM NO. 3 *
****

***** END OF FILE *****

****
* SUMMARY FOR THIS RUN *
****

COUN. NO.	N	M	A	B	FI	FO	FOX	FOO	LREST	BEST COST	NO. OF LEV.	TIME(CS)
1	3	2	1000	1	3	3	3	3	4	12024	4	310
2	3	2	1000	1	2	2	2	2	6	******	6	1363

Fig. 5-5(t)

6. INTERMEDIATE RESULTS FOR UNFINISHED JOBS

In each computer run, the execution time allowed for each problem cannot exceed the limit specified on the < specification card >. When the specified time is to expire but the problem currently under execution is not finished, then the intermediate results are punched on the cards so that the user can use these cards to run this problem next time.

During the execution of a problem, the computation time, TIME, spent so far will be compared with the specified limit, TLIM, each time after the initial network is obtained or after the transformation or the transduction is applied on a network. If the comparison satisfies one of the following three inequalities*, then this problem will be executed continually. Otherwise the intermediate results will be prepared.

```
TIME +   2 sec <TLIM     if n = 3
TIME +  20 sec <TLIM     if n = 4
TIME + 120 sec <TLIM     if n = 5
```

The punched deck for the intermediate results consists of types (i), (ii), (iii), (iv), (v) and (vi) cards as well as other two new types of cards. We will explain this by giving an example later. Fig. 6-1(a) shows the information about the problem specification for the 5-variable parity function. The control sequence used is OPT2 and the computation time specified is 126 sec†.

---

*We make this comparison so that we will not stop the execution too late (in the case of 5-variable functions, some procedures need more than one minute to do the transduction) and we have enough time to do the punch and the print.

†This means that the execution is terminated when TIME + 120 > 126, or when TIME > 6 sec.

The problem is not finished because the specified time is not enough. So some necessary information is printed in Fig. 6-1(b), the intermediate results are punched and the system goes to read another problem. Since only one problem is included, the summary table is printed and then the execution is terminated. The information shown in Fig. 6-1(b) is for rerunning the program and can be ignored by the user. (In Fig. 6-1(b), six asterisks are printed under the column entitled with "BEST COST", because no result which satisifes the given restrictions is obtained yet).

The punched deck contains the following cards:

(1) The first four fields of the < specification card > are specified as 1, since we prepare the < heading card >, the < problem parameter card >, the < output function card >s and the < control sequence card >s for each unfinished problem. The last two fields contain the same values as the original problem; i.e., the same values for PUNC and TLIM. The user may change the value of TLIM before rerunning the program.

(2) The < heading card > usually has the following format:

OLD PROB. # X , FROM INITIAL NETWORK XXXX, INTERMEDIATE RESULTS

(3) The values of the parameters, except the last two fields, on the < problem parameter card > are the same as the original problem. Parameter NEPMAX has the value 2^{N-1} if this field was blank in the original problem; and parameter RERUN has the value 1 to indicate that this is not the first time that

```
**********************
*  PROBLEM NO.   1   *
**********************
```

*** FIVE-VARIABLE PARITY FUNCTION ***

NUMBER OF VARIABLES = 5
NUMBER OF FUNCTIONS = 1
COST COEFFICIENT A = 1000
B = 1

--- UNCOMPLEMENTED VARIABLES X ---

FAN-IN = 4
FAN-OUT = 4
FAN-OUT FOR EX. VARIABLES = 4
FAN-OUT FOR OUTPUT GATES = 4
NO. OF LEVELS = 100
TIME SPECIFIED FOR THE PROBLEM = 12600 CS

CONTROL SEQUENCE SPECIFIED BY THE USER

OPT2

FUNCTION NO. 1
01101001100101101001011001101001

Fig. 6-1(a)

```
**********************************************************************************************************
* TIME TO BE EXPIRED, THIS PROB. WAS STOPPED AUTOMATICALLY.  USE THE PUNCHED DECK FOR CONTINUING THE JOB. *
**********************************************************************************************************
```

DUMP OF PARAMETERS

IL	STAGE2	STAGE4	ITER	STEP1	STEP2	STEP3	STEP4	BSTCST	TIME	LEVLIM	OPTION	FIFOS
2	2	0	1	1	1	0	0	2000000	723	100	2	1

```
**********************
*  PROBLEM NO.    2  *
**********************
```

***** END OF FILE *****


```
*****************************
*   SUMMARY FOR THIS RUN    *
*****************************
```

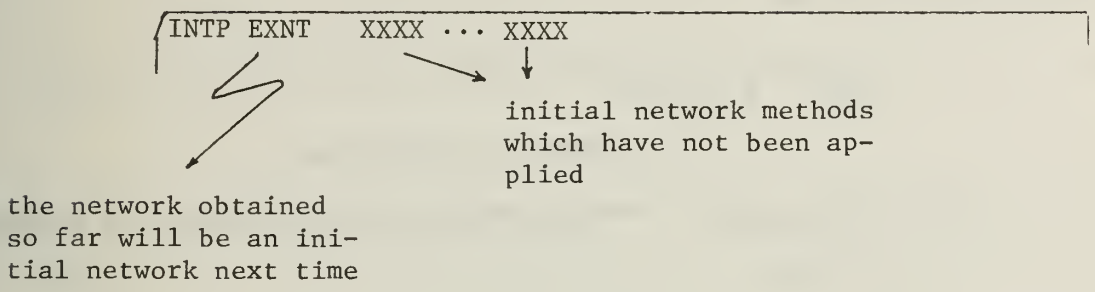
COUN. NO.	N	M	A	B	FI	FO	FOX	FOO	LREST	BEST COST	NO. OF LEV.	TIME(CS)
1	5	1	1000	1	4	4	4	4	100	******	100	723

Fig. 6-1(b)

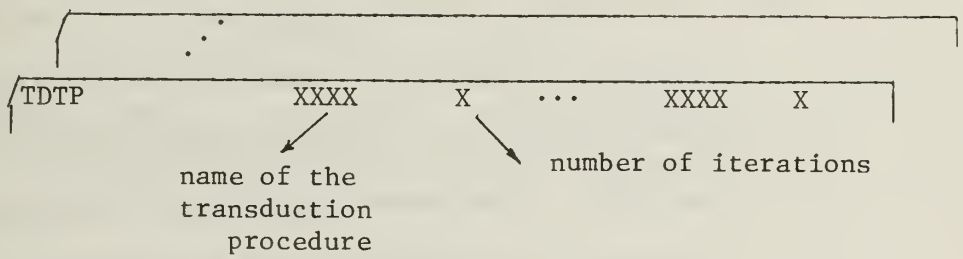
the problem will be run.

(4) The < control sequence card >s are divided into five sets of cards (usually five cards).

The first set of cards, consisting of one card, contains the names of the initial network methods which were specified for the original problem but have not been applied. It has the following format:

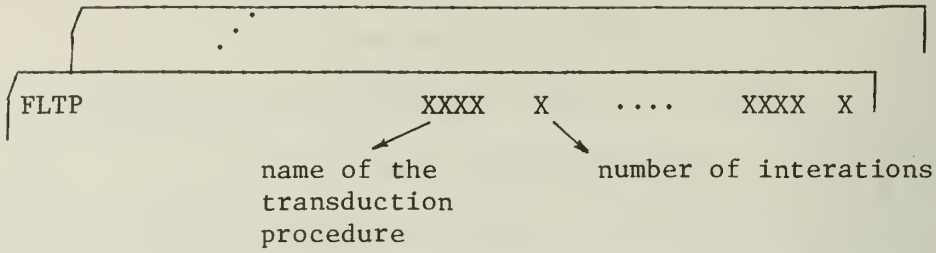


The second set of cards (usually one card is enough) corresponds to the specification of the transduction procedures under no fan-in/fan-out restrictions and no level restriction. The corresponding format is shown below.

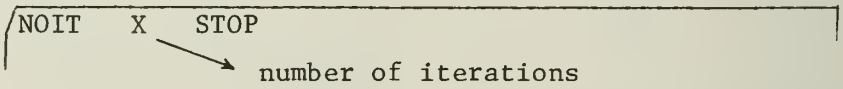


The third set of cards have only one card. It has 4 characters, JEFF, only. If "JEFF" is not specified on the control sequence cards in the original problem, then this card is not punched.

The fourth set of cards correspond to the fan-in/fan-out restricted and/or level-restricted transduction steps. The corresponding format is shown below.



The fifth set of cards consist of only one card. It contains the number of iterations that the TT-sequence to be applied. The corresponding format is shown below.



(5) The < output function card >s are the same as the original problem.

(6) The < current status card > is a new type of card which contains the information about the status of the current problem. This information will be used for rerunning the problem.

(7) The < connection-description card >s for the network which which was just processed when the problem stopped have the same format as the < connection-description card >s for the input data. The network described by these cards will be treated as the initial network for the next run.

(8) The < connection-description card >s for the best network obtained so far are also punched (if there is no network satisfying the given restrictions, then these cards do not exist).

This network will also be read in and stored so that we can ke the "real" best network at the end of the processes.

Fig. 6-2 gives the punched deck for the example shown in Fig. 6-1. Notice that there are no < connection-description card >s for the best network since no network which satisfies the given restrictions has been obtained. Another thing which should be mentioned is that the character "#" which was specified in the original < control sequence card >s is now replaced by 99, i.e., the selected transduction procedure or the TT-sequence will be executed at most 99 times. Apparently this is equivalent to executing it repeatedly until there is no further improvement in the cost.

```
  1   1   1   1   0 126
OLD PROB. #  1, FROM INITIAL NETWORK  BANDB   .   INTERMEDIATE RESULTS
  5   11000   1   X   4   4   4   4 100   0   0  1  16
INTP    EXNT
TDTP    NTG3      99    NTG1      99
JEFF
FLTP    NTE1      99
NOIT    99  STOP
01101001100101101001011001101001
  2   1   0   1   1   1   0   0 2000000     626100  0  1
 1 2 913   2 3 4 5 627   3 5 6192123   4X3   5X214   6X11525   9 3111615212426
11X2X418  13X315161921  14X115  15X1X3202227  16X2X3232425  18X4X52526  19X41820
26  20X1X3X41826  21X5182224  22X1X3X51824  23X2X4  24X2X3X526  25X1X224  26X2X3
X427  27X1X2X4X5  **
```

1. < specification card >
2. < heading card >
3. < problem parameter card >
4. < control sequence card >s
5. < output function card >s
6. < current status card >
7. < connection description card >s for the current network

Fig. 6-2 Punched deck for the unfinished job in Fig. 6-1

REFERENCES

[1] Hu, K.C. and S. Muroga, "NOR(NAND) Network Transduction System, (The Principle of NETTRA System)," Report No. UIUCDCS-R-77-885, Department of Computer Science, University of Illinois, Urbana, Illinois.

[2] Hu, K.C., "Level-Restricted NOR Network Transduction Programs," Report No. UIUCDCS-R-77-849, Department of Computer Science, University of Illinois, Urbana, Illinois.

[3] Su, Y.H. and C. W. Nam, "Computer-Aided Synthesis of Multiple Output Multi-Level NAND Networks with Fan-In and Fan-Out Constraints," IEEE Trans. Comput., Vol. C-20, December, 1971.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-77-887	2.	3. Recipient's Accession No.
4. Title and Subtitle PROGRAM MANUAL: FOR THE NETTRA SYSTEM			5. Report Date August, 1977
			6.
7. Author(s) K. C. Hu			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. NSF DCR 73-03421
12. Sponsoring Organization Name and Address National Science Foundation 1800 G Street, N.W. Washington, D.C. 20550			13. Type of Report & Period Covered Technical
			14.

15. Supplementary Notes

16. Abstracts

This is the program manual for the NETTRA system which can design near-optimal, multi-level and loop-free NOR(NAND) networks under fan-in/fan-out restriction and/or level restriction. Given function(s) may be completely or incompletely specified and both complemented and uncomplemented external variables are permitted as inputs. The user can specify the control sequence (the types of the initial network methods and the types and the order of the transduction procedures to be applied) to solve his problem. Besides, four control sequences are provided for the users who are not interested in the details of how to specify the control sequence. Facilities for treating unfinished jobs due to the expiration of computation time are also provided by the system.

17. Key Words and Document Analysis. 17a. Descriptors

Logic design, logic circuits, logic elements, programs (computers).

17b. Identifiers/Open-Ended Terms

Computer-aided design, transduction procedures, transformations, near-optimal networks, optimal networks, fan-in, fan-out, level, NOR, NAND, NETTRA system, integer programming logic design, program manual.

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release Unlimited	UNCLASSIFIED	
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70)

USCOMM-DC 40329-P71